TEC-0115

Task-Driven Active Vision for Security and Surveillance

H. Keith Nishihara, et al.

Teleos Research
2465 Latham Street, Suite 101
Mountain View, CA 94040

August 1998

Approved for public release; distribution is unlimited.

Prepared for:
Defense Advanced Research Projects Agency
3701 North Fairfax Drive
Arlington, VA 22203-1714

Monitored by:
U.S. Army Corps of Engineers
Topographic Engineering Center
7701 Telegraph Road
Alexandria, VA 22315-3864

US Army Corps
of Engineers
Topographic
Engineering Center

TEC

U.S. ARMY TOPOGRAPHIC ENGINEERING CENTER
TEC

Destroy this report when no longer needed.
Do not return it to the originator.

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

The citation in this report of trade names of commercially available products does not constitute official endorsement or approval of the use of such products.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
August 1998

3. REPORT TYPE AND DATES COVERED
Technical October 1993-September 1996

4. TITLE AND SUBTITLE
Task-Driven Active Vision for Security and Surveillance

5. FUNDING NUMBERS
DACA76-93-C-0017

6. AUTHOR(S)
H. Keith Nishihara, J. Brian Burns, Rick Marks,
Stanley J. Rosenschein, Phil Kahn, Stan Birchfield,
Dan Perrin, Joe Digiovanni, Dave Badtke, David J. Beymer

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Teleos Research
2465 Latham Street, Suite 101
Mountain View, CA 94040

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Defense Advanced Research Projects Agency
3701 North Fairfax Drive, Arlington, VA 22203-1714

U.S. Army Topographic Engineering Center
7701 Telegraph Road, Alexandria, VA 22315-3864

19. SPONSORING / MONITORING AGENCY REPORT NUMBER
TEC-0115

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*
This report details progress that Teleos Research has made in the development of computer vision and visual attention mechanisms for the support of Security and Surveillance tasks. Theories and experimental results are presented in five related areas: figure-ground discrimination, tracking servos, object recognition based on parts extraction, object recognition based on local geometric structure, and object modeling. The report also discusses trends toward real-time software implementations of this type of vision technology on commodity processors.

14. SUBJECT TERMS
Security and Surveillance, Active Vision, Figure-Ground Discrimination, Object Recognition

15. NUMBER OF PAGES
89

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT
UNCLASSIFIED

18. SECURITY CLASSIFICATION OF THIS PAGE
UNCLASSIFIED

19. SECURITY CLASSIFICATION OF ABSTRACT
UNCLASSIFIED

20. LIMITATION OF ABSTRACT
UNLIMITED

Contents

List of Figures

PREFACE

This research is sponsored by the Defense Advanced Research Projects Agency (DARPA) and monitored by the U.S. Army Topographic Engineering Center (TEC), under Contract DACA76-93-C-0017, titled, "Task-Driven Active Vision for Security and Surveillance." The DARPA point of contact is Dr. Ed Carapezza, and the TEC Contracting Officer's Representative is Mr. Thomas Hay.

1 Introduction

This is the final report on work accomplished by Teleos Research on a three-year contract supported by DARPA's Real-Time Planning and Control Program.

This project focused on the study and implementation of practical, task-driven computer vision applied to security and surveillance tasks. A guiding principle in this work has been the notion that the sophisticated performance observed in biological vision systems is to a large degree derived from the fluent use of simple and robust measurement capabilities.[1] The task at hand drives how and when sensory actions are to be performed, but the sensory measurements themselves can be generic. A major objective of this research effort has been to identify and develop basic visual measurement mechanisms that can be employed flexibly in diverse applications requiring visual perception. This report describes progress in four areas relevant to the development of active visual perception capabilities:

1. Attentional mechanisms. Methods for detecting likely targets in a video image stream from a camera that is possibly moving
2. Smooth real-time tracking. Methods for maintaining smooth pursuit of a patch of surface over short periods of time
3. Reacquisition and classification. Methods for determining whether a detected visual object is of interest (e.g. a human versus a cat), and for discriminating amongst multiple similar targets to stay with the same one, especially across occlusions
4. Object modeling. Methods for measuring properties of a tracked visual object, such as its size or pose.

Figure 1 illustrates the interrelationship between these component visual tasks.

1.1 The Security and Surveillance task domain

A visually rich, but otherwise restricted, application domain is vital for guiding and evaluating a core research effort of this type. Surveillance and Security (S&S) provides a rich research context in which to test task-directed vision techniques. It supports perceptual tasks of interestingly different types over a broad range of difficulty. These include detection of new or reappearing objects, classification of movement patterns, detection of common destinations, detection and tracking of motion in visual or IR imagery with an active head, and discrimination of humans using size, shape, color, texture, and motion cues. S&S problems often require visual strategies in order to perform well, and goal-directed attentional mechanisms are key in all but simple cases.

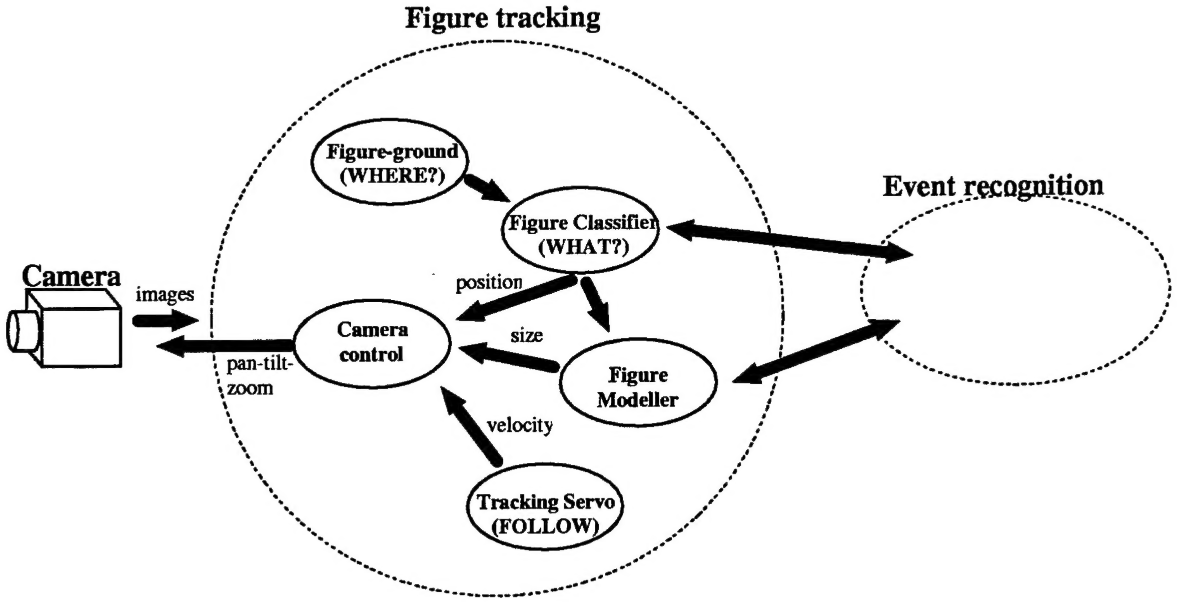


Figure 1: Event perception is a process that naturally modularizes into tracking and interpretation modules. Tracking persistent visual objects, such as walking humans, requires competences of several types as shown in this figure. This research program has made contributions in each of these areas.

A key capability required to increase the level of automation in the above applications is the ability to automatically, rapidly, and consistently recognize objects and events observed under natural viewing conditions. This requirement is made difficult by the real-time nature of the task and the complexity of finding and analyzing object images when they are undergoing articulated motions under varying lighting, and against complex, possibly moving backgrounds. Vision-based technologies have previously been insufficient in meeting these demands.

Modular real-time active vision measurement capabilities can be applied to a large set of task-oriented systems. Hard perception problems are easier to solve with a good set of primitive measurement capabilities. To be of practical value, a visual measurement capability must provide robust and appropriate measurements in time to be useful at a cost that is not prohibitive. One consequence of these considerations has been an increased focus on demand-driven visual measurements as opposed to earlier approaches that attempted to carry out a full scene analysis prior to making any use of the information derived. That latter approach had the disadvantage of being computationally expensive and did not easily support the differing and sometimes dynamically changing needs of behaving perceptual systems. The work reported here places emphasis on identifying and making local measurements appropriate for visual tracking applications.

1.2 Military Significance

In both the public and private sectors there is increasing demand for systems that take advantage of on-line sensor data, especially real-time video data. In many of these applications, special-purpose sensors, coupled with structured work environments, have made it possible to deploy versions of working systems. Despite these early successes, however, the economics of deployment remains unfavorable. In the security and surveillance domain, the ability to do quick detection and classification of objects can add value to systems monitoring an interior area for intruders or performing outdoor perimeter control. Current-generation motion-detection systems are hampered by their inability to recognize or classify the objects causing the motion. The coupled detection, tracking, and recognition techniques reported in this report would immediately increase the value of intrusion-detection systems and reduce the personnel needed to man them.

Computer perception applied in the security and surveillance domain has a wide range of immediate governmental and commercial applications that include: law-enforcement and security (e.g., detection of public criminal activities, such as drug dealing on street corners, building surveillance, detection of loitering and parking lot security), nuclear storage warehouse security, and consumer mobility pattern analysis in a store (e.g., to detect shoplifters, and to optimize product placement for consumer traffic patterns). Current automated security systems are prone to high false alarm rates and often the only acceptable solutions require direct monitoring by human personnel.

2 Figure-ground discrimination

Fast detection of object candidates (*figures*) in images prior to recognition is an important first step towards accomplishing many visual tasks, especially in dynamic time-critical conditions. It allows attentional control to focus processing resources more efficiently. It also can improve the performance of subsequent image analysis activities.

By isolating a figure significantly smaller than the whole image, the resources of the recognition process can be employed more efficiently. More processing can be performed in the area likely to contain an object of interest. Two important examples of this in biological vision systems are motion and color-based figure-ground discrimination. Each of these modalities is used as an attentional cue that draws fixation reflexes to anomalous areas in the visual field. Both types of biological figure-ground systems are remarkable in that they operate robustly against confounding stimuli. In the case of motion, abberant motions against a moving background can be detected. With color, figures can be detected against natural shading illumination variations that can significantly affect the physical spectral distribution at the eye.

An important characteristic of these biological systems is their *parallel* nature. They operate over a wide visual field and yield *popout* percepts wherever they detect an anomalous image area.

The color and motion modalities have the nice property of requiring fairly local computations. They also can operate usefully with very simple models of what constitutes a figure, for example, anything different from the nominal background in their respective image representations.

Teleos has investigated both of these modalities for implementing fast, wide-field figure detection mechanisms for use in figure tracking. Each has yielded figure detection algorithms that can be implemented efficiently in software, and individually performs well enough to guide an active camera. The following sections describe these algorithms and characterize their performance envelopes.

2.1 Positive and Negative Criteria

The task of detecting figures moving against a background can be accomplished using positive information about the figure, such as its color or texture, or particulars of its motion. For example, in an earlier tracking system based on the Prism-3 system[2] we used binocular stereo to accomplish a coarse figure-ground discrimination, and then used optical flow measured from textures near the center of the detected figure to guide camera movement. This system worked well, but relied on two-camera stereo to provide a robust measure of a quality of the figure not shared by the background, namely, its distance from the cameras.

Many positive criteria, such as similarity measures using color or texture, can be used to find figures once a model is obtained for the characteristic color or texture of the figures of interest. Conversely, they are more difficult to use to accomplish the initial detection.

Another approach to finding figures is to look for consistencies in the background that can be used to classify background portions of the image. Image areas that do not fit the background model could be deduced to be possible figure locations. This latter approach can be thought of as using negative criteria to detect figures. Under this research program we have investigated the use of background motion for this purpose.

There are several advantages to using motion as a negative criteria. First, background motion is much easier to model than are the motions associated with a complex articulating figure. Second, objects moving faster than the vision system's ability to measure can still be detected as long as the background velocity can be handled. Third, in some instances, negative criteria algorithms require less computation and the underlying theories are simpler.

A primary weakness associated with negative criteria-based techniques is their inability to discriminate one figure from another. This leads to confusion anytime the tracked figure gets near another object. Positive criteria methods are, by their nature, often able to make finer discriminations between multiple figures.

2.2 Motion-based figure detection

One of the strongest perceptual cues to the presence of a person in a visual scene is motion. The challenge to using this cue is the preponderance of natural motion in an image. To be effective, the detection scheme must isolate motion characteristics that can be associated with the objects of interest. To study negative criteria, we consider situations in which figures cover a relatively small percentage of the camera field of view and move against the background which is generally moving coherently. The assumption of uniform background motion has proven to be an effective criteria for discerning figures in motion fields.

It is possible to use image differencing to detect areas of figure motion, but this requires either the use of static cameras [3, 4], or some means of precisely measuring the background image motion. This can be accomplished using camera positional feedback [5], but this technique requires accurate, synchronized sensors which are often not available, and it restricts the camera's motion to rotation about its focal point.

2.2.1 The discorrelation at dominant motion algorithm

After exploring a range of alternative methods for detecting figures using motion, the following discorrelation at dominant motion (DDM) algorithm was developed. Its approach is to model the background motion field using correlation techniques. Once this motion is known, a final correlation is performed registering the current image with a previous image that has been warped to null out the background motion. Thus, any figures with differential motion relative to the background pop out as poorly correlated regions in this comparison step. The locations of these regions are measured and reported as detected figures.

The background motion model must be estimated in the presence of confounding figure motions. Two methods have been developed to accomplish this. Both rely on the assumption that any figures constitute a minority of the total image area. The first method makes a further simplifying assumption that the background motion can be approximated as a translation. In this method, a dense optical flow image is computed and the resulting velocity vectors are histogrammed to identify a dominant translational motion. The second method employs whole image correlation to identify the dominant velocity in the image. In this case it is assumed that the figure motion will not significantly affect the correlation peak position since the figure occupies a small percentage of the image area. This second approach can be enhanced somewhat by masking out regions where figures had previously been detected.

Figure 2 shows an example of a complex articulating figure (a person) extracted from the background via motion analysis. These frames are from a large sequence in which the DDM algorithm was employed successfully.

This basic design of the DDM algorithm has been further enhanced to allow the detection of very slowly moving objects by freezing the reference image, and updating it only when the motion between the reference and current live image become too large to handle. Another enhancement maintains a representation of figure position when no figure motion is occurring. Also, the figure map is now represented at a higher resolution for a more detailed understanding of the object's extent in the image. Figure 3 shows an example of a detected region associated with a person moving relative to the background given the current version of the system.

This algorithm has been tested on video sequences of subjects walking in an indoor environment. It also has been interfaced to a commercial pan-tilt-zoom security camera and was used to demonstrate active camera control in tracking people in indoor and outdoor settings.

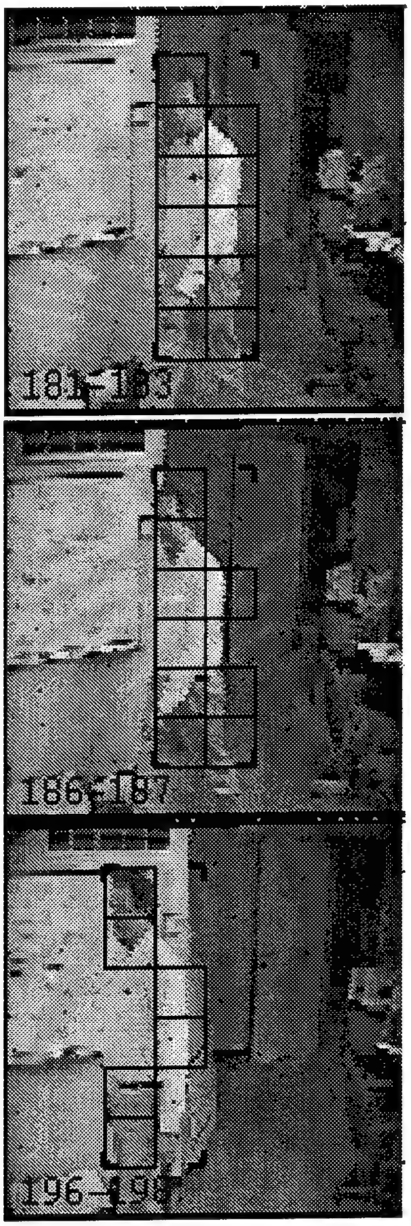


Figure 2: Frames from a video sequence showing movement-based detection of an object (black squares). This *negative criteria* figure-ground discrimination algorithm allows detection of moving bodies against a moving background as occurs when the camera is in motion.

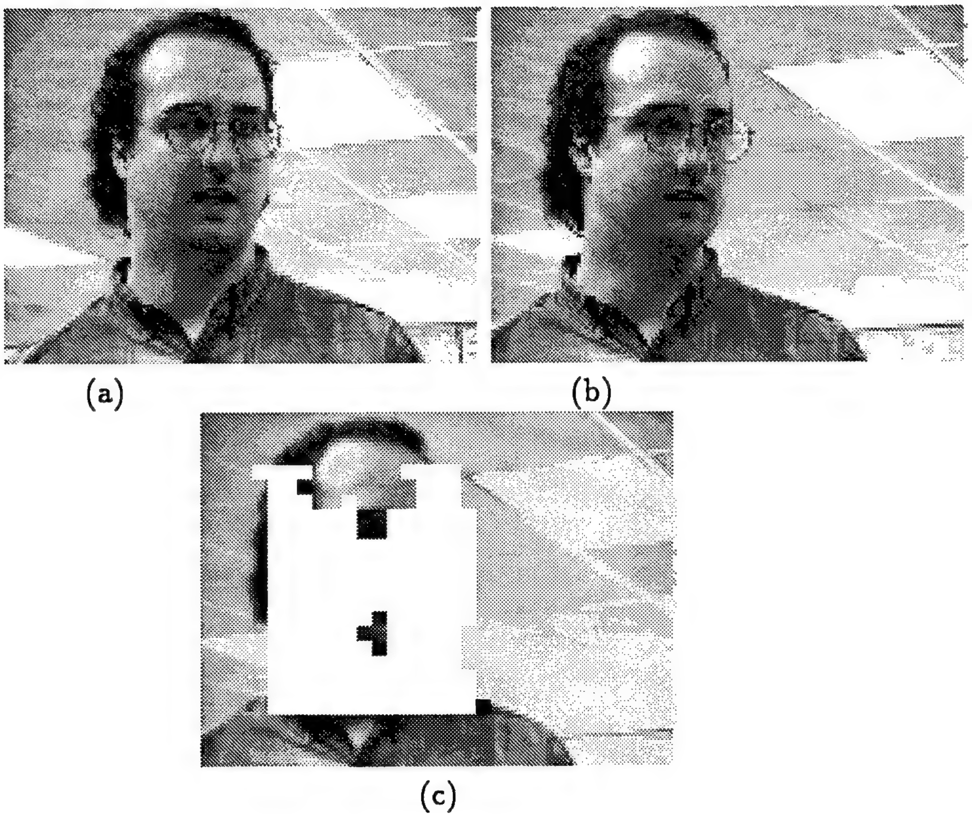


Figure 3: Example of a detected region associated with a person moving relative to the background. (a) First frame, (b) second frame, and (c) detected region shown in white. Note background motion caused by camera pan.

2.2.2 Performance evaluation

The DDM algorithm works sufficiently well to allow detection and sustained tracking of a single moving figure using an active camera head. The current version has been used to competently track people in real-time for periods of up to a half an hour (54,000 frames), and in the presence of multiple moving objects for up to twenty minutes (36,000 frames). The motions of the tracked subject can be quite large, covering the natural range, and also quite complex, including rapid 3-D rotations. Also, the system has been tested for over a dozen subjects and many different backgrounds.

The algorithm's primary failure modes uncovered in testing were of three principal types: insufficient background texture, isolated background motions that depart from the overall background motion model, and situations where the pure translation model was violated.

Backgrounds with very low texture contrast occur in some S&S environments such as office environments with solid colored walls and white boards. In these situations the background velocity cannot be estimated using the histogramming or correlation techniques described above. This actually is not a problem for truly textureless backgrounds since an error in the background velocity model has no consequence for the final discorrelation detection stage of the algorithm. It is a problem for many intermediary situations where background texture is present, but is insufficient to support the computation of a reliable background velocity. In such cases, an error in the background velocity estimate results in textured background regions being labelled as figure locations.

The second problem category includes a range of image events that mimic desired figure motions. Some examples include moving shadows on walls, wind-blown shrubbery outside of windows, flashes of light on walls reflected from passing vehicles, and even moire patterns on window blinds caused by aliasing with the sensor array.

2.3 Color-based figure detection

Color is an excellent candidate for a positive-criteria approach to detecting figures since desired figures often have colors distinctive from the background. As a positive-criteria method, it requires a stronger model of the target figure in order to detect such figures. Swain[6] developed a robust algorithm for locating and tracking brightly colored cereal boxes in a real-time active vision system. We explored the use of some of his ideas for detecting human faces and torsos given an initial color model constructed using a negative criteria mechanism like the DDM algorithm described earlier. Recently, facial color has been proposed as a primary criteria for tracking systems, due in part to the convenient discovery that normalized skin color is relatively stable for people of different complexions[7, 4].

2.3.1 Histogram Backprojection algorithm (HB)

Swain's histogram backprojection algorithm[6] scans the entire image for regions that have colors consistent with a model following a weighting scheme that gives preference to colors that are present in the figure model, but are not common in the background. The result is a *figure map* that indicates locations with colors consistent with the figure model. Peaks in a smoothed version of this map are flagged as possible figures.

2.3.2 Performance evaluation

The HB algorithm is fast since it does not require computations over extended image neighborhoods. Colors, especially flesh tones from human targets, are distinctive enough to allow detection and tracking, using color matching information alone over extended periods. This makes it an effective tool for accomplishing simple reacquisition of tracked targets once a color model is built.

The major failure modes of this figure-ground mechanism are rapid changes in the spectral content of the subject illumination. It also is susceptible to confusion with backgrounds with colors similar to those used in the target model. The use of facial color also fails when the face turns away from the camera.

These occurrences are not correlated closely with the common failure modes of the DDM negative criteria algorithm described in section 2.2.1. Thus, a combination of the two figure-ground techniques has a wider operating range.

2.3.3 Normalized color representations

The basic HB algorithm was extended by normalizing the color components with respect to brightness before the histogramming step. In this study, the raw image data from the camera was in the form of Y, U and V components, where Y is related to the brightness of the color, and U and V are distinct differences in red, green and blue.

The red, green and blue color components are monotonic functions of the amount of the light shining on the projected scene surface. For example, the red component of an image of a red apple will increase in magnitude if the amount of light on the apple increases, as long as this light has power in the red part of the spectrum (e.g., white light). Thus, these components are not invariant to change in total amount of illumination. Since Y, U and V are all linear functions of red, green and blue, they also are not invariant to change in the amount of light.

As a tracked subject walks from one part of a room to another—or into another room entirely—the amount of light falling on the subject generally changes greatly. Thus, it is important to attempt to normalize (stabilize) the features used to detect and localize the subject. In the experiments discussed above, histograms of raw (Y, U, V) were used. In this section, the U and V components were first normalized. Since Y, U and V are all linear functions of red, green and blue, dividing U and V by Y will tend to cancel changes because of illumination intensity. This was done for colors with Y above some minimum value, which was set to 30. For colors with Y below 30, U and V were both set to 0.

Histograms of size 16 by 16 of the normalized colors (U', V') were compared with histograms of (U, V), as subjects moved about in front of the camera. The peaks of dominant color were observed to shift in both histograms as the subject illumination changed. One suspect for these shifts was the white balance in the camera. This mechanism compensates for the variation in size of a color region by shifting the colors of all regions. Thus, when a subject moves and changes the sizes of color regions, all the values shift. After turning off the white balance in the camera, the shifting of the histogram peaks became greatly reduced.

Without the white balance, the shift in the normalized colors did appear to be less than that of the unnormalized colors. Shifts of about four histogram cells were observed to be reduced to shifts of about two cells after normalization. However, the normalized colors did still shift, possibly because of changes in the color of the light hitting the subject. In the experiments performed here, the subject was near a computer monitor and received light from ceiling lamps. The monitor was distinctly bluer than the ceiling lamps, and the subject was walking around within 2 feet to 5 feet from the monitor.

In summary, some reduction in lighting sensitivity seems to be happening, however a change in lighting color is a practical reality; the simple method used here does not compensate for it. Overall, the tracking behavior seemed unchanged by the normalization. Since the normalization involved a division at every pixel, in may not be worth the computation. Further experiments on the tracking behavior are required to determine this.

3 Tracking servo

The figure-ground mechanisms described in Section 2 yield a figure map indicating image locations with higher probabilities of being occupied by a visual object of the desired type. These *figure* regions typically do not have well-defined boundaries. For example, the motion-based DDM algorithm *sees* disparate motion that may arise from the whole body of a human figure, or just from a moving arm or head. Thus, the resulting figure regions can fluctuate significantly in size and center location causing a simple camera pointing algorithm to behave erratically. Other problems occur when one figure comes in close proximity to another. This is especially bad with negative criteria techniques that have no means for discriminating between separate figures.

One means for stabilizing the temporal behavior of a figure-ground guided tracker is to employ a motion continuity constraint. A simple form of this requires that the figure location not accelerate too quickly. This helps to dampen out some of the positional noise because of segmentation instability in the figure-ground processing. This instability is caused by non-uniform and changing patterns of discorrelation across the figure's extent. When the segmentation is viewed in real time on a monitor, it appears as a flickering effect.

Stabilization and tracking responsiveness can be improved more significantly by measuring the local image velocity over the tracked figure. By following the integral of local velocity measurements, some of the positional instability caused by figure segmentation instability and proximity with other figures can be eliminated.

3.1 Patch tracking algorithm

Frame-to-frame area correlation can be used to measure local image motion of a single patch of surface. When centered on a figure under track, this technique has been found to be effective for maintaining a responsive lock on that figure. This technique gives precise measurements of movement and allows a PID control system to move the camera servo motors in unison with the figure. Equally important, it keeps the cameras from moving when the figure does not, but the figure-ground mechanisms see a momentary change in discorrelation pattern, which shifts its measured center of mass.

We use area correlation on the Sign of Laplacian of Gaussian (SLOG) filtered images.[8] SLOG correlation provides a robust measure of image motion even under degraded contrast conditions. For patch tracking, a relatively large correlation window is employed to improve the correlation peak shape—from 20 to 60 pixels square. We currently employ a window on the smaller end of this range. Subpixel interpolation of the correlation peak position typically gives image velocity resolution on the order of 0.1 to 0.3 pixels.

A useful property of SLOG correlation is the broadness of the correlation peak that is proportional to the size of the LOG filter employed. This broad peak allows the algorithm to tolerate a larger range of non-translational image deformations. Surface motions that have some non-translational variation generate a peak at the average velocity under the correlation window. This aids the patch tracker as well as the negative criteria DDM algorithm described earlier. These non-uniformities in the patch motion arise from small rotations and expansions of the target as well as from deformations of the target shape.

The patch tracking algorithm correlates a single patch taken from the prior image with patches over a 2-D search area on the current image. The required search range is determined by the anticipated target velocity and the measurement rate of the patch tracker. If the measurement rate is 30 Hz and image velocities as large as 200 pixels per second are to be handled, then a search range of about 7 pixels in all directions must be covered. Note that the area to be searched increases with the square of the interval between successive frames.

The effective tracking rate of the system can be linked to a limit on the object's maximum acceleration by using the previous image velocity as a predictor for the location in the new image to search about. With the same 7 pixel search radius and 30 Hz measurement rate, as in the previous example, this would allow accelerations as large as $6300 pixels/second^2$.

3.2 Performance evaluation

As intended, the patch tracking algorithm can provide very responsive tracking of a target once its image location is known, provided that it exhibits stable texture markings. Most human targets exhibit sufficient texture to drive the patch tracker.

The primary failure mode observed has been the loss of the figure center because of 3-D rotation of the figure. This causes the patch tracker to follow the surface texture over the figure's occluding boundary. This allows the correlation patch to pick up background texture and become locked to it rather than the figure.

A combination of patch tracking control with positive and/or negative figure-ground modules makes it possible to restrict the occurance of this *sliding off* phenomena while preserving the responsive characteristics of patch tracking. The location of the background edges also can be used to realign the tracker[3]. Another possibility is to realign the object's contour with motion discontinuities[9]. In section 7 we will present a model based technique that employs an elliptical head model to keep the tracker from drifting away from the target object.

4 Multiscale ridge and blob centers

Once candidate figures have been detected by attentional mechanisms like those described in the previous sections, it is important to determine their identity. Several types of recognition are relevant for tracking people for security and surveillance. One is to determine whether or not the candidate figure is the same as one tracked in previous frames. We call this the reacquisition problem. Often object descriptions as simple as a color histogram will be sufficient to accomplish this task. The second type of recognition capability is object classification. In this case figures located in the visual field must be classified according to what they are. This may be a coarse categorization into human and non-human forms or it can involve more precise identification of individuals. In this section and the next, methods for representing image structure for the purpose of these various forms of recognition are presented.

In this section the concept of image representation in terms of local centers is motivated, and computational models of the concept are compared. A particular model, the *appropriate-scale ridge*, is developed and demonstrated. In this model, local centers are defined as smoothed image extrema that also are maximal, with respect to scale, in the magnitude of the second spatial derivatives. The basic ideas are extended to color and texture data, and the texture features are demonstrated via face detection.

A major goal of the research presented in this section is the recognition of natural objects given natural viewing conditions. An important example of this problem is the visual detection, identification, and understanding of people. The recognition of people requires the extraction of visual information that tends to be stable with respect to such transformations as limb articulation and variations in clothing. Given this, image properties that are potentially useful for recognition include aspects of the proportions and geometric arrangement of visually distinct object parts. Thus, our basic approach to recognition is to extract key visual parts, model objects as relational graph descriptions over these parts, and recognition of the objects by matching images to the graph descriptions. Since graph relations can be arbitrary, the relational graph formalism is sufficiently general to include approaches to rigid object matching, such as alignment-based methods[10], as well as recognition strategies for more variable configurations of parts. By analyzing partial graph matches that can be efficiently searched, yet provide strong indications of object presence, parts-based and graph-based recognition can be made fast and effective [11].

Parts-based recognition has been proposed at least as early as [12], and is in contrast with, current image-based methods for people detection and recognition [13, 14]. To detect people in all possible articulations, views and lighting, image-based approaches would require a large collection of reference images. In addition, Bichsel and Pentland observe that since the set of images being sampled is highly non-convex, its linear approximation (e.g., eigenimages) has very limited effectiveness for reliable detection. Given this, parts-based recognition appears more promising.

This section describes a computational model of detectable visual parts that are a potential foundation for parts-based recognition. In general, such parts should be visually stable and informative in the sense that lower order relations over them provide discriminating properties for natural objects.

Given these objectives it appears difficult to formulate a model of appropriate visual parts in terms of either local image edges or 2-D connected components. A set of extracted edges generally provides too low a level of representation for recognition: single edges usually represent little information; matching large numbers of edges can be very slow; and considerable effort may be required to produce more compact representations from a set of edges. Edge level representations may be useful in situations where a single, rigid reference frame can be used to evaluate the registration of fragmented data [10]; however, such a context is often not available in natural object recognition.

Image connected components generated by pixel classification (image segmentation) tend to have problems with stability: small changes in view or lighting can produce dramatic changes in the topology of the connections. This, in turn, produces dramatic changes in the size, shape, and position of major components. In addition, complex assemblages of parts are often represented by a single component, requiring potentially expensive shape analysis to extract the salient parts.

Instead of edges and 2-D connected components, we define salient image parts in terms of *local centers*: visually compact regions that have significant internal-external value contrast in any of various image measurements, such as brightness, color, texture, depth or motion. Given this model, local centers are extracted that optimize internal-external contrast with respect to position and scale of the center; the local centers can then be organized into basic visual parts via simple geometric rules. Local centers need not be mutually exclusive: centers of different sizes, representing significant structures at different scales, can be associated with the same location in the image.

The advantages of this scheme over an edges-only model are that (1) the basic unit of representation already encodes something about the position, size and internal characteristics of a visually distinct part of the object, and (2) the extracted centers potentially require less organizational processing and selection than edges to be useful for recognition. The advantage of this scheme over 2-D connected components is that the basic units are not as complex and unstable. Figures 4 through 7 show examples of objects that can be readily and fruitfully represented as sets of local centers of brightness value (Figures 4 and 6), and local centers of texture orientation and magnitude (Figure 7). The local center representation need not be used to the exclusion of extracted edges, which might further localize the boundaries between regions of complex shape. However, we believe that explicit edge information is not always reliable, or required, and should often be organized with respect to the extracted local centers.

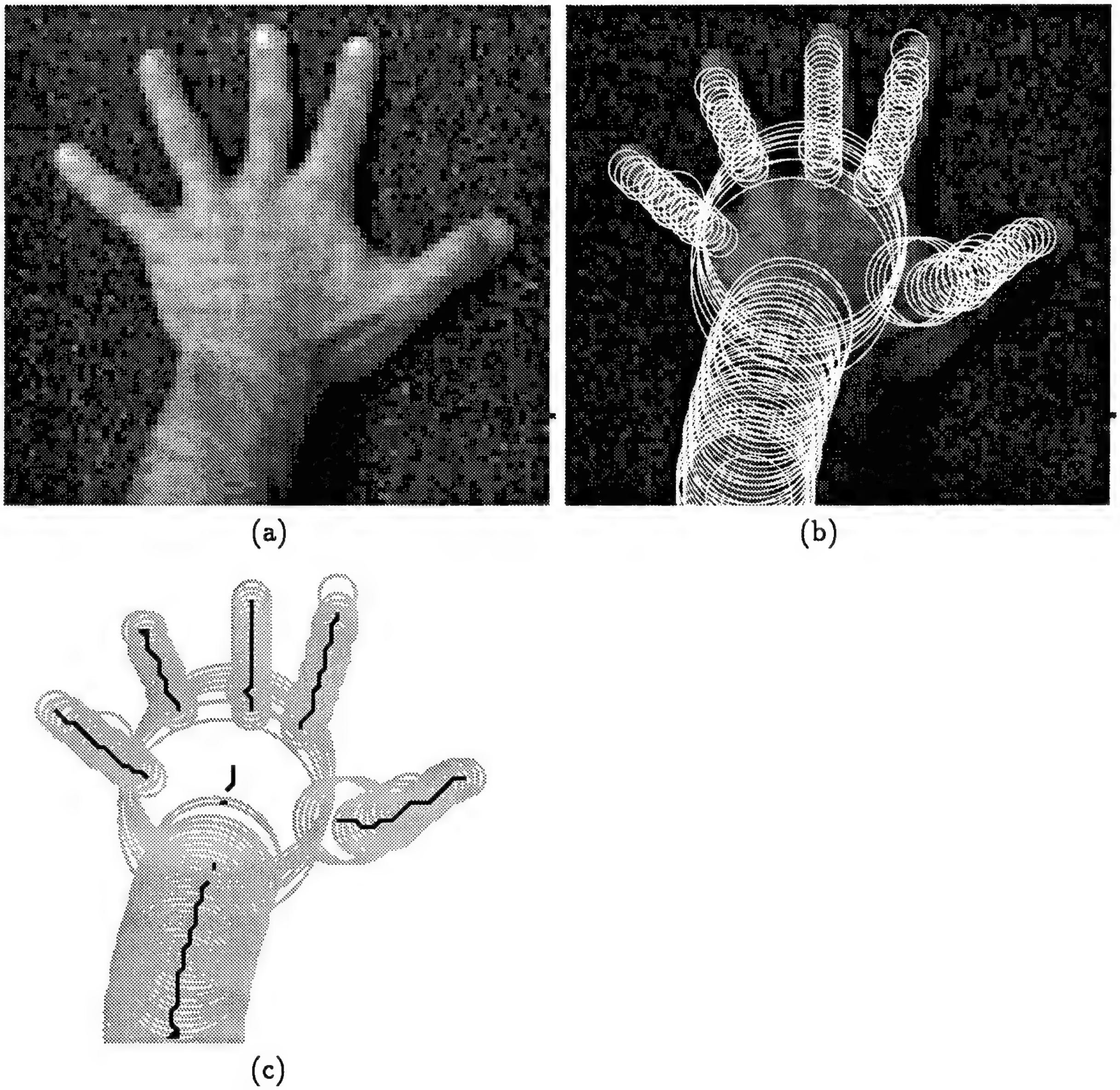


Figure 4: An example of a natural object readily represented as a set of *local centers*, visually compact regions that have significant internal-external contrast. The local centers shown have been extracted using the appropriate-scale ridge model discussed in the text: (a) image, (b) scale and position of detected centers represented by radius and center of circles, and (c) centers linked into parts.

4.1 Models of multi-scale local centers

This section presents computational models of local centers in terms of intensity contrast; later sections will expand the discussion to include color and texture.

The basic models can be characterized as medial axis transforms, Laplacian of Gaussian peaks and ridges, intensity peaks and ridges, and what is referred to here as *appropriate-scale ridges.* The latter model combines the concept of intensity ridge with local contrast maximization across scale to effectively localize, in both image position and scale.

4.1.1 Medial axis transforms

The basic idea of reducing complex shapes to a set of local centers has been proposed at least as early as the medial axis transform of [15, 16]. The transform reduces a shape to the set of circles of varying position and scale that are multi-tangent with the shape. Since these circles lie along the axes of shape parts, they can be used as a basis of parts extraction and representation. However, the basic transform is very sensitive to small perturbations of the shape boundary, thus, multi-resolution versions have been developed [17, 18]. Unfortunately, the basic idea still requires that an intact region be extracted prior to analysis. Also, the extraction of shape parts is not a function of the local contrast of the parts themselves. Both of these properties limit the usefulness of the idea as a general method of representing visually salient image structures.

4.1.2 Peaks and ridges of LoG

One way to more directly associate local centers with places of local maximum internal-external contrast is to define them as the peaks and ridges of the image convolved with the Laplacian of the Gaussian (LoG)[19][1]. The peaks of the difference of low-pass filter output is essentially the same idea[20]. This model also is related to the bar detector designed by Canny. In [21], the image is convolved with a mask that has a cross-section, well approximated by the second spatial derivative of the Gaussian. Peaks in the magnitude of the output are then extracted.

In the LoG model, an image structure is represented by a collection of LoG peaks and ridges at different scales. The LoG operator at a given scale and image position is simply the difference between the weighted average of a central zone of width related to the scale and the weighted average of a surrounding zone. Thus, peaks in the output clearly represent contrast maxima with respect to image position. Given the Gaussian filter, the weights define a smooth envelope over each zone, and the output extrema exhibit

[1] For this discussion, we will refer to both peaks and pits as *peaks*, and ridges and valleys as *ridges.*

stability with respect to significant image transformations[19]. In addition, the number and positions of the extrema at different scales vary as a function of some interesting shape attributes, such as the number of significant bumps on the shape and the pointedness of the bumps. Thus, the set of peaks and ridges of the LoG output generated at multiple scales may provide a useful representation for shape discrimination.

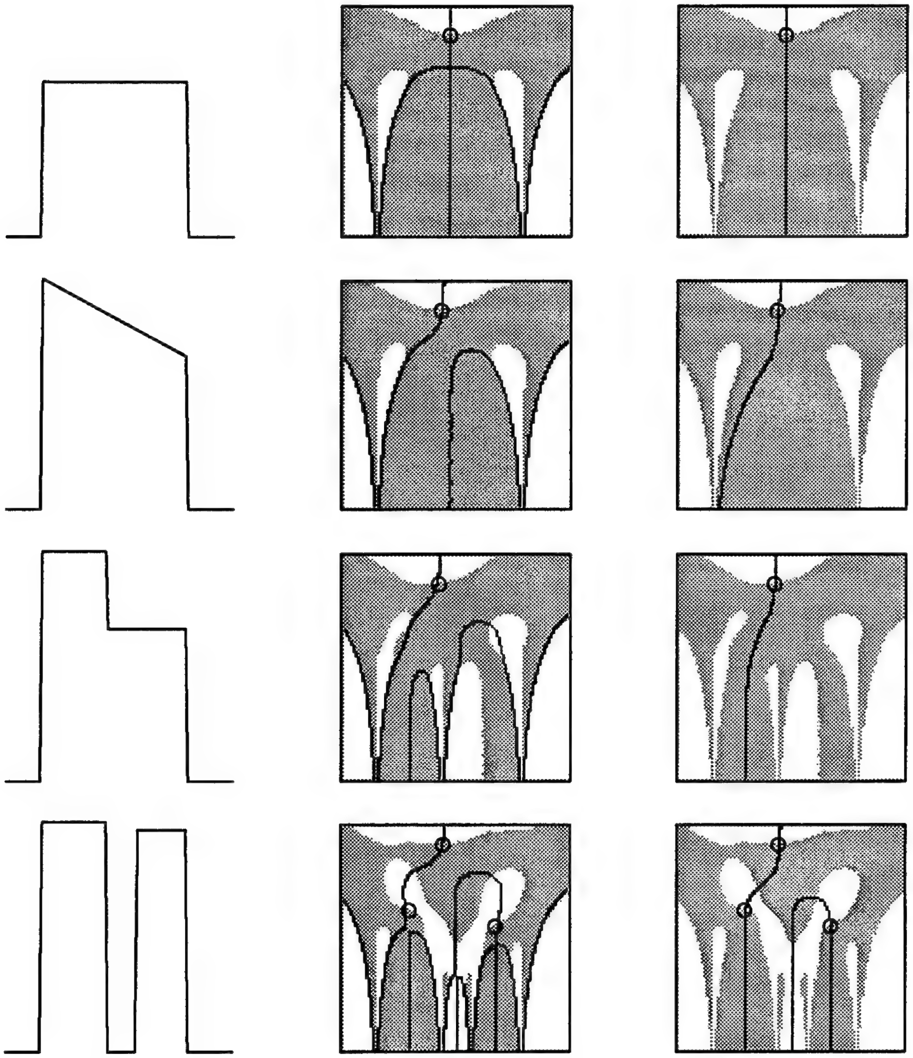


Figure 5: Scale-space analysis of G_{xx} versus appropriate-scale peaks (see text for discussion). First column: 1D images (x by intensity); second column: x by $\log\sigma$ plots of $G_{xx\sigma}$ sign bits (white/gray = $+/-$) with G_{xxx} zero-crossings (black); third column: same sign bit plot with G_x zero-crossings (black). Circles mark position and scale of desired local centers.

One property of the LoG peaks and ridges is their sensitivity to intensity edges or shoulders. Basically, local extrema appear at image positions where the center region is at one side of an abrupt intensity change and a maximal portion of the surrounding region is on the other side. This edge response happens at every scale where the edge is present, can be of high magnitude, and has a position that shifts as a function of the scale.

The objective of the present study is to extract local centers that are defined as compact *regions* of significant internal-external contrast. Given this objective, an extracted local center should represent the overall positions and extents of an object and its distinct parts, thus, provides a compact, hierarchical decomposition of complex object shapes. This objective could be satisfied by a LoG peak-and-ridge model if the edge responses could be suppressed or labeled. This has been proposed by Crowley and Parker[20]; however, it is not generally practical. The LoG peaks, with respect to image position (x, y), can be caused by whole regions (local centers) or edges. The difference between the two types is that the former peaks also are local extrema with respect to the Gaussian scale parameter σ, while the latter are not. Thus, the suppression of edge response is essentially equivalent to finding LoG local extrema with respect to σ as well as image position (x, y). Given most practical situations, only a discrete sampling of σ is available and true peak detection is difficult.

This can be understood by considering the equivalent 1D case: detecting peaks in $G_{xx}(\sigma) * I$ with respect to (x, σ), where $G_{xx}(\sigma) * I$ is the convolution of 1D image I with the second derivative of the Gaussian at scale σ. These peaks are at points (x, σ), where zero-crossings of $G_{xxx}(\sigma) * I$ and $G_{xx\sigma}(\sigma) * I$ coincide. In order to select only true peaks with respect to both σ and x, the zero-crossings of $G_{xxx}(\sigma) * I$ and $G_{xx\sigma}(\sigma) * I$ must be far enough apart from each other in the neighborhood of edges to not appear coincidental given discrete sampling methods. The examples in the second column of Figure 5 show that this misleading coincidence at edges may often be hard to avoid in practice. The first column shows 1D images (x by intensity) and the second column shows the scale-space plots [22] of $G_{xxx}(\sigma) * I$ and $G_{xx\sigma}(\sigma) * I$ (x by $\log \sigma$) in response to each image. The gray represents areas of negative $G_{xx\sigma}(\sigma) * I$, while white is positive (the borders of the two areas are the zero-crossings). The thick black lines are zero-crossings of $G_{xxx}(\sigma) * I$. True peaks in $G_{xx}(\sigma) * I$ with respect to (x, σ) should be near the desired local centers, represented as circles in the plots.

As can be seen, there are zero-crossings of $G_{xxx}(\sigma) * I$ and $G_{xx\sigma}(\sigma) * I$ that *appear* coincidental in the finite sampling of σ, are not near the true peaks and are instead associated with edges. This occurs even with the one hundred samples of σ used in this example, which is generally not practical. Tracking the zero-crossing curves through scale-space and selecting points of extreme $G_{xx}(\sigma) * I$ with respect to a parameterization of the curves might accomplish the desired effect, though this would seem to be a complex process, sensitive to sampling and noise. It is likely to be even more difficult to analyze the zero-crossing *surfaces* in (x, y, σ).

Given the objective of the current study, and the difficulties of using a straightforward LoG peak-and-ridge scheme as a model of local centers, an alternative method has been developed. However, the patterns of LoG peaks across scales still may be a useful means of uniquely characterizing shapes for their discrimination once they have been extracted via a local center process.

4.1.3 Peaks and ridges of intensity

In Subirana-Vilanova and Sung[23], the problem of an edge response is avoided by a more complex set of operators. The bar detection operator of Canny[21] is "split" into two complex assymetric operators that are combined by selecting the minimum magnitude response of the two at every point. The resulting non-linear operator is applied everywhere in the image at different directions and scales, and the maximum output at each point is selected. The resulting output surface is then analyzed for skeletons, much like systems of peaks and ridges. The method does appear to avoid edge responses; however, it is not clear why the process prior to the peak and ridge extraction is necessary. Directly detecting peaks and ridges in smoothed images may work as well, or better. The two contributions to local center analysis made in [23] are that the scale of the underlying image structure is also recovered and the operator has been extended to color images. We believe that these two aspects of the problem can be handled without requiring such a complicated system. This is discussed in the next sections.

A potentially simpler way of ensuring that the *centers* of visually distinct concentrations in value are detected, instead of the edges, is to use the peaks and ridges of smoothed intensity. They are always associated with the centers of some relatively brighter or darker region at a given scale[24]. By extracting intensity extrema over a range of scales, a compact description of the image related to local centers may be possible. This basic idea has been studied in [24], with related ideas developed by Koenderink[25], and Lindeberg[26]. The concept is consistent with the observation in Morrone and Owens[27], that edge-like and bar-like (i.e., ridge or local center) image features can be distinguished by the relative magnitude of the outputs of odd versus even operators (phase). They studied 1D operators that closely resembled G_x (odd) and G_{xx} (even) at a fixed scale. The centers of the bars were at points where the odd function is zero, and not the even function, and vice versa for centers of edges. This is equivalent to detecting ridges at the zero-crossings of G_x. Results in Gauch and Pizer[24] and Lindeberg[26] show that detected intensity peaks and ridges correspond to intensity regions and curvilinear features.

In Gauch and Pizer[24], extrema detected at one scale are related to those at other scales via a scale-space tracking process, though their methods seem complex and tentative. Given tracked extrema, the scale of the underlying image structure has been either defined as the scale at which the associated extrema are annihilated[26] or defined as a *set* of positions and scales[24]. For intensity ridges, the latter output is in the form of a 2-D sheet in scale-space.

It seems problematic to define scale in terms of the annihilation point, since this event is a function of structures external to the local center or region being tracked. For example, the annihilation will occur at a much smaller scale if the region is between two near and large regions than if it is between two far and small ones. Likewise, the complex 2-D sheet representations seem unwieldy and only indirectly related to the underlying properties of width, position and contrast of a given region. Given the complexity and indirection of methods based strictly on intensity extrema, it is useful to consider other possibilities.

4.1.4 Appropriate-scale ridges

We have developed a model of the local center concept called the *appropriate-scale ridge*, which emphasizes the best of both the LoG and intensity ridge approaches. Like the LoG model, internal-external contrast determines the scale and saliency of a local center, and, like the intensity model, the position of a local center is constrained to being at an intensity peak or ridge.

This approach is best understood in the 1D intensity case. The third column of Figure 5 shows some examples for 1D images of appropriate-scale ridge points. First, consider the example in the first row: a 1D box-shaped region of some width that is bright relative to its surround. Over a range of scales σ, the position $x_{\max}$ of the maximum of $G(\sigma) * I$ with respect to image position x will correspond to the centroid of the region, where $G(\sigma) * I$ is the convolution of image I with the Gaussian at scale σ. Thus, this extremum is a good model of the associated local center's position. The local center is visually significant if its internal-external contrast is high enough. In 1D, a useful center-surround contrast operator at a given scale σ is simply $G_{xx}(\sigma) * I$, where $G_{xx}(\sigma)$ is the second derivative of the Gaussian with respect to x. Thus, the local center contrast is high if the magnitude of $G_{xx}(\sigma) * I$ is high at $x_{\max}$. In addition, the scale σ that best corresponds to the width of the underlying region is the one for which the magnitude of $G_{xx}(\sigma) * I$ at $x_{\max}$ is maximal.

In summary, this implies the following useful model of a 1D local center. A point (x, σ) is an appropriate-scale peak if it is:

1. At a local extremum, with respect to image position x, of $G(\sigma) * I$, and
2. At a local extremum, with respect to scale σ, of $G_{xx}(\sigma) * I$. (The senses should also be compatible: a minimum of $G_{xx}(\sigma) * I$ if $G(\sigma) * I$ is at a maximum, and vice versa.)

Points that satisfy the above conditions are at intersections of the zero-crossings of $G_x(\sigma) * I$ and $G_{xx\sigma}(\sigma) * I$ in scale-space (x, σ). Specifically, they are at the intersections where the sign change for $G_x(\sigma) * I$ in the direction of positive x is the opposite as the sign change for $G_{xx\sigma}(\sigma) * I$ in the direction of positive σ. The zero-crossings for each image in the first column have been plotted (x by $\log \sigma$) in the last column of Figure 5. The gray represents areas of negative $G_{xx\sigma}(\sigma) * I$, while white is positive (the borders of the two areas are the zero-crossings). The thick black lines are zero-crossings of $G_x(\sigma) * I$. The circles represent the position and scale (x, σ) of desired local centers in the image.

As can be seen, the examples include 1D regions with asymmetric bumps and gaps of significant magnitude. In each case, zero-crossings of compatible sign change coincide at points that are near the centroid and width of the actual regions of significant contrast (i.e., no edge effects); elsewhere, the compatible zero-crossings are well separated. Thus, in 1D, the appropriate-scale peak model appears to correspond to our concept of a local center.

4.2 Appropriate-scale ridges in 2-D

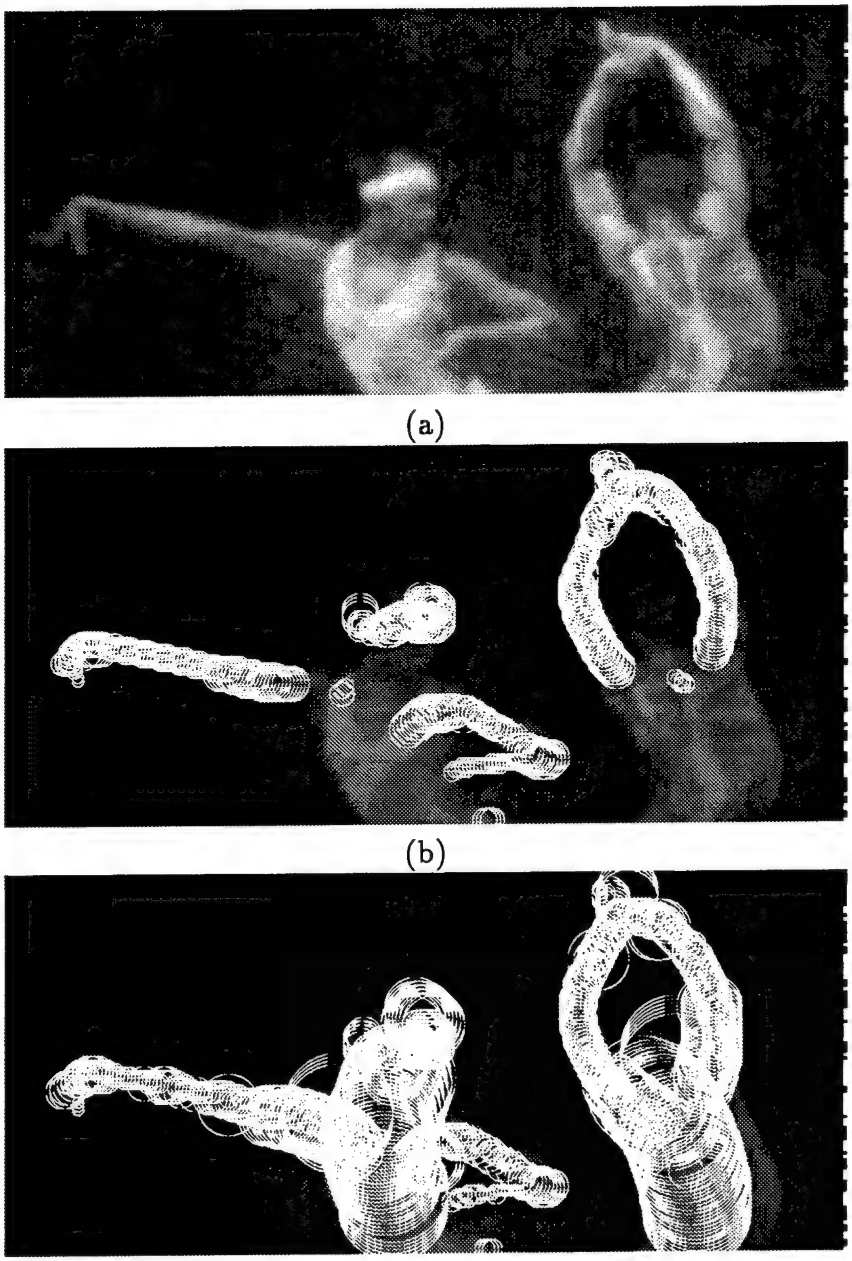
A version for 2-D images (x, y) can be readily defined in terms of the second directional derivatives of $G(\sigma) * I$, or $G_{tt}(\sigma) * I$ for image position parameter t, varying in some image direction θ with respect to the x axis. The direction of the maximum magnitude second directional derivative at (x, y, σ) corresponds to the direction of maximum internal-external contrast at this point and scale; the positional parameter in this direction will be referred to as $t_{\max}$. Given this, the following model is used. An image point and scale (x, y, σ) is an appropriate-scale ridge point if it is:

1. At an extremum in $G(\sigma) * I$, with respect to $t_{\max}$, and
2. At an extremum, with respect to σ, in $G_{t_{\max}t_{\max}}(\sigma) * I$, the local contrast at (x, y, σ). (Again, the senses should also be compatible: a minimum of $G_{t_{\max}t_{\max}}(\sigma) * I$ if $G(\sigma) * I$ is at a maximum, and vice versa.)

In order to compare the local contrast measured at different scales σ, the volume defined by the operator surface in its negative (center) and positive (surround) parts must be made constant with respect to σ. This can be achieved by multiplying the operator by σ^2.

For the discrete implementation of this model, σ is sampled exponentially.[25] For each sampled scale σ_i, 2-D image ridge points are detected using rule (1) above, and a ridge point (x, y, σ_i) is selected if its local contrast exceeds the local contrast at (x, y, σ_{i+1}) and (x, y, σ_{i-1}).

Figures 4 and 6 show the results of applying this process to images of natural objects with parts of various positions and widths. In each example, a local center is shown as a circle width radius and center corresponding to the local center scale and position. Only the bright centers are shown in each case; the dark centers define the shape of the complementary regions. In Figure 4(b), output is shown from five scales, and a contrast threshold of 10 was used. Note that the widths seems appropriate for the underlying substructures (up to the scale sampling) and the parts are densely sampled along their axes. Figure 4(c) shows the extraction of salient object parts by simply linking neighboring centers into chains. Figure 6 shows the results on human figures. In



(c)

Figure 6: Appropriate-scale ridges of human figures: (a) image, (b) ridge output from four scales, and (c) ridge output from five scales.

Figure 6(b), output at four scales is shown (from finger to arm widths), and Figure 6(c) shows 5 scales (from finger to trunk width). Note that the operator avoided producing edge responses, and the overall shapes of these complexly shaded figures are reasonably represented by the general configuration and widths of the local centers.

4.3 Color model

It is useful to expand the concept of an appropriate-scale intensity ridge to multimodal input. The use of multiple channels of input increases the likelihood that a given local center of contrast is associated with a figure that should be considered distinct from its surround: the more ways in which the internal data differs from the external, the more likely the internal region is of a different material and scene object. Color provides such multiple channels. Also, given typical scene lighting, color components such as hue are relatively insensitive to internal variations in the object projection because of shading. Thus, with color contrast, the overall shape of the figure outline (e.g., someone's shirt or pants) is often emphasized more than the details (e.g., the cloth folds and wrinkles).

A color image is a mapping from two parameters (x, y) to a vector of color components (u, v, w). Since it is a parameterization of a 2-D manifold in the color space, there are no "peaks" or "ridges" in the sense defined for intensity images. However, a simple extension of the ridge or peak concept can be used. Consider a 1D image, $I : x \mapsto (u, v, w)$. Regions of relatively constant color in the image correspond to sections of the manifold where the parametrization "slows down"; in other words, the arclength of the parametrization $s(t)$ is relatively small. Consistent with this, image color edges are places where $s(t)$ is relatively large. Thus, we can construct an analogous concept to the intensity case by defining a color peak as a point of local minimum arclength $s(t)$, and an appropriate-scale one as a peak that is also a maximum with respect to smoothing scale σ in the second derivative of $s(t)$ with respect to t.

For a 2-D image, $I : (x, y) \mapsto (u, v, w)$, there is an analogous concept of appropriate-scale color ridge. This has been developed and implemented. It has been given a tentative testing and evaluation on a full-body color image of a person (not shown here). All limbs and the torso were extracted from the background, each as a separate body part, with approximate, but reasonable estimates for the part proportions.

4.4 Texture and face detection

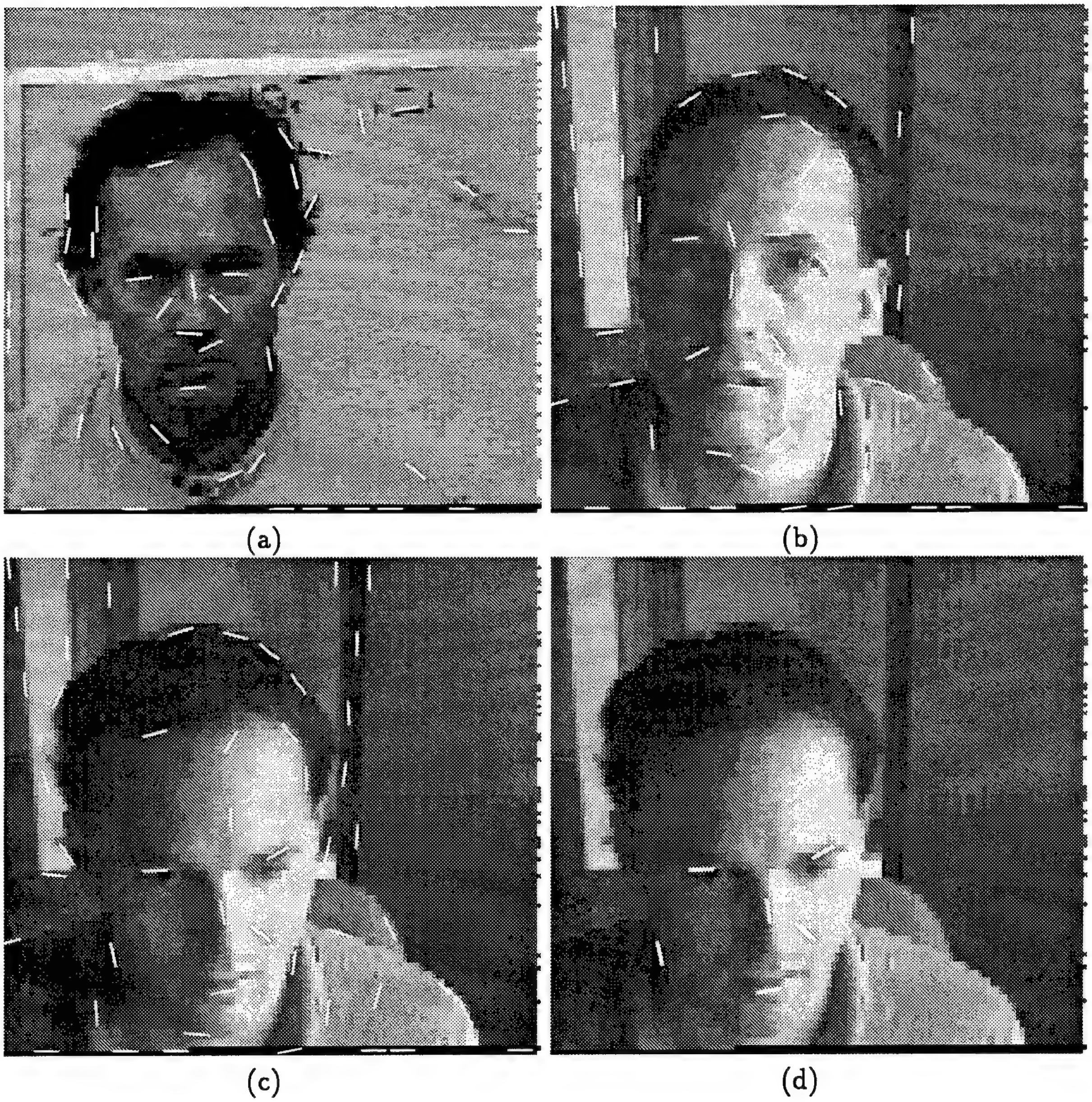
Sometimes a figure's internal shading variation is also important for parts extraction. For example, salient features of an object surface, such as a nose on a face, often appear as a patch of image shading or texture. To recognize faces, it is useful to extract local centers that correspond to concentrations of certain shading or texture attributes. For this reason, the ideas of Gaussian smoothing and peak detection also have been applied to local centers of texture and shading properties; however, currently only a single-scale version has been implemented and tested.

Many aspects of a local intensity patch vary considerably as the light source or camera angle changes. One aspect appears relatively stable and was used as the basis for face representation: the dominant orientation of the intensity variation in an image patch relative to the face's vertical axis. The magnitude of the variation of the intensity in a patch can be different for different directions of measurement. The variation can be measured in terms of the first, second or other derivatives of the image function. By defining the orientation of a texture or shading patch as the dominant orientation of the intensity variation, and by measuring the orientation with respect to the object's reference frame, we have a feature that is relatively stable with respect to lighting and camera change. The salient features of the human face (e.g., the nose, mouth, eye and cheek regions) generally have an expected, dominant orientations, thus, local centers of texture orientation and magnitude could be useful features to extract for face detection.

A type of texture feature was developed that is related to this idea, though others could be used. It is based on the fact that smoothing a texture field suppresses the resulting texture magnitude in areas without coherent texture orientation, at the given scale, while keeping the magnitude of coherent textures constant. The effect is to produce peaks in magnitude at points with texture that is relatively coherent or contrasting in orientation from the surround. The specific model is defined as the following steps:

- An oriented texture field (dx, dy) is computed by detecting ridges in the LoG output at some scale and using the ridge direction as the texture orientation (Modulo 180 degrees: ridges of similar orientation but different sign are considered parallel. This is accomplished by multiplying the vector angle by two.)
- The texture field is then smoothed by convolving the individual components dx and dy by a Gaussian.
- The peaks in the resulting texture magnitude $|dx, dy|$ above some threshold are selected as the texture tokens; the orientation of the smoothed field at each peak point is used as the token orientation.

This scheme was tested on a set of face images with different lighting and head positions, holding operating parameters constant. Figures 7(a-c) show example results, where peaks are drawn as bars. In general, the detection and localization seems reasonably stable. The output was evaluated by using it to detect faces. Detection was accomplished by searching for the correct 2-D similarity transformation between each image and a 2-D face model constructed of similar features. Candidate transformations were generated by assigning pairs of model tokens to pairs of image tokens and computing the model-to-image transformation that best aligned them. Each candidate match was evaluated



(a)

(b)

(c)

(d)

Figure 7: Local centers of texture orientation: (a-c) examples of detected texture centers (bars) show the stability under variation in view and lighting, (d) the automatically generated face model match to the third image. In (d), the centers with highest confidence face feature labels are shown. Additional centers were labeled, and all were labeled correctly.

by checking the alignment of the rest of the model tokens to the nearest image tokens. The face detector was tested on a 20 frame video sequence of a 3-D-rotating head. Faces were correctly matched in 16 out of 20 of frames; see for example, Figure 7(d). The incorrect matches were filtered out via smooth motion modeling of the head. With the match score threshold set so that 13 of the correct matches are selected, no false matches are selected in the 20-frame sequence and only 7 out of 100 false matches were accepted in a sequence of 100 frames panning a cluttered scene without faces.

5 Consensus-based recognition

The previous section described methods for locating image objects at different scales using global characteristics, such as concentrations of mass in a given image feature. In this section we explore the use of more detailed local geometric patterns for use in recognition.

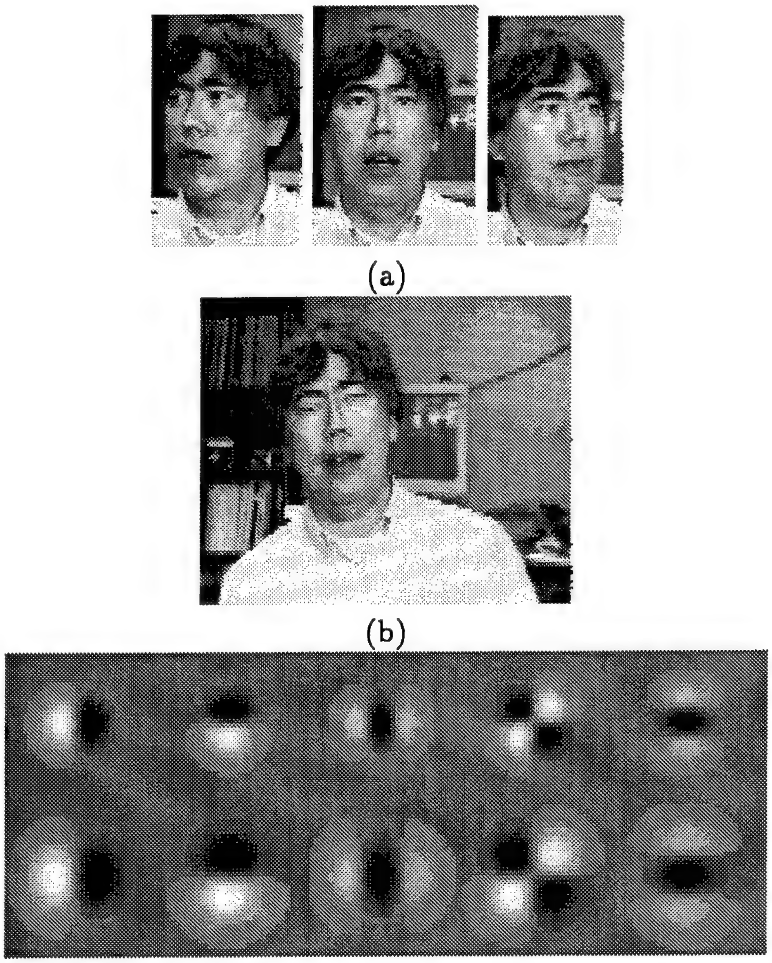
More specifically we descibe the fruitful combination of two useful methods in recognition: consensus or voting-based approaches and moment-based representations. The basic idea is first demonstrated using moments of the image brightness on the detection of 3D objects undergoing 6D variations in position and orientation (*pose*). The idea is then extended to handle large variations in light source using moments of local texture orientation. This idea also is demonstrated on real image data.

5.1 Consensus methods

In consensus-based recognition, correspondences between localized parts of an object model and localized parts of the input image are formed, and each such local match votes for the object and object poses that are consistent with it. Detection occurs if there is a large enough accumulation of votes for some object and pose. This is the basic approach associated with generalized Hough transforms [28] and geometric hashing [29], both of which have been used to detect 3-D objects. The advantages of these techniques are simplicity and robustness with respect to large corruptions or loss of data such as by occlusion. Current 3D recognition designs based on this approach are limited either by dependence on special, potentially difficult-to-detect local image features, such as line junctions [28], [29] and elliptical arcs, or image features that are too indistinct and ubiquitous, such as edge points. In the latter case, the system can be plagued by too many local feature correspondences to process (millions or billions), and too much clutter in the space or hash table in which the vote cluster detection is performed. Methods of detection by voting become much more efficient if the local features are more uniquely characterized by higher dimensional descriptions [30], and the process is more broadly applicable if the local feature representation used is more general than the detection of specific structures (such as line junctions and ellipses).

5.2 Moment representation

Ideally, local feature representation would be a simple function of the original image data and the whole object would be modeled by processing images from distinct views, as in Figure 8. In the moment-based approach to recognition, the image is filtered by a set of 2-D functions that represent or are related to the moments of the brightness distribution



(a)

(b)

(c)

Figure 8: Moments based on derivatives of Gaussian smoothed image patches provide a relatively informative representation of the image that can be normalized to minimize effects of changes in contrast and some image distortions. Tests of a view-based approach using a moment representation is illustrated here. (a) Image samples representing the subject from distinct views. (b) A sample from a 150 frame sequence of the subject talking, blinking and rotating, used to test the recognition system. (c) Derivatives of the 2-D Gaussian at different scales: the bottom row is one half-octave larger in scale than the top, and the derivatives are (left-to-right) g_x, g_y, g_{xx}, g_{xy} and g_{yy}.

in the image. These techniques include traditional moment methods [31] and [32], as well as current methods using steerable filters ([33], [34], [35]), and are related to techniques using Gabor wavelets [36]. The steerable filters currently used are the derivatives of the Gaussian taken at various scales [37], which are actually linear combinations of the brightness moments weighted by Gaussians at given scales [38].

Figure 8 shows all of the first and second derivatives of the Gaussian for two different scales. The bottom row shows derivatives at a scale that is larger than the top row by $\sqrt{2}$. What are being shown in the figure are the convolution kernals associated with the moments: the light regions are positive coefficients of the kernals, and the dark regions are negative. The distribution of the coefficients define the position, scale and aspects of the intensity pattern that the moment is tuned to. By using multiple moments centered at some position, the intensity pattern of that locale can be represented. Moments provide a relatively informative representation of the image that can be normalized with respect to changes in contrast and some image distortions. They are a simple, general basis for representation, requiring only sample images of the object to generate a model (Figure 8). In addition, if a relatively small series of moments are used for each local patch, the image processing and matching of the moment features can be made fast on standard processors.

One shortcoming of moment matching is the stability of the moments with respect to occlusions, image clutter and other disruptions. Because of this, moments have been traditionally used to detect objects that are isolated and often in silhouette form [31], [32]. Recently, Rao and Ballard [34] have used methods of robust feature vector matching to improve this, going far to demonstrate the discrimination potential of moments from a single patch. However, their design is still very sensitive to certain disruptions, it accepts too many false positives, and it may have high storage costs. To get the large number of Gaussian derivatives (45) required for their robust method, the image patch encoding uses nine derivatives measured over five octaves of scale. If the encoding is centered and scaled so that the image support for the largest scale is largely within the boundary of the object (i.e., extraneous data has low impact), then the support region for the smallest scale occupies only 1/256th of the object's region, and three-fifths of the features have support regions that are only one-sixteenth of the object region. This should make the system very sensitive to occlusion or other changes in a small, central area of the image patch.

Also, with only forty-five measurements and the very tolerant match acceptance required for robustness, the moments of a single image patch alone have a false positive rate that can be improved on: multiple false points in a single image are said to have at least 0.9 correlation with a given reference patch, and a seventy percent discrimination rate across a set of rigid objects is reported. Finally, Rao and Ballard's robust feature methods may also incur an efficiency penalty: to handle random perturbations of a forty-five component moment vector, the feature indexing system may have to store a lot of the vector variations.

Instead of depending on measurements from a single image patch to represent and detect an object, it seems better, in terms of occlusion insensitivity and processing complexity, to use localized measurements from multiple patches distributed about the object's image and of different sizes. The multiple matches of these individual patches can be used to detect and localize an object via the consensus or voting methods discussed above. This potentially fruitful combination of methods is stressed and demonstrated here: the approaches of moment representation and vote clustering are complementary.

The use of multiple, local moment-based matches also is a basis for the method of [35]; however, they only use a few, manually selected local patches (five) and no uniform method of determining large clusters of consistent matches. Rao and Ballard [34] also discuss using multiple patches, but again, only in a sparse sampling and without a method of detection by integrating multiple match results under arbitrary variations in 3-D position and orientation. A larger number of locally encoded patches is considered in [36], but this work emphasizes an iterative match refinement technique. Their method is potentially useful for verifying and improving the pose estimate after the object has been roughly detected and located.

Our basic idea of local moment matching and global voting-based cluster detection has been applied to moments of brightness and local texture orientation. The latter shows promise in the presence of large, complex changes in lighting.

5.3 Brightness moments

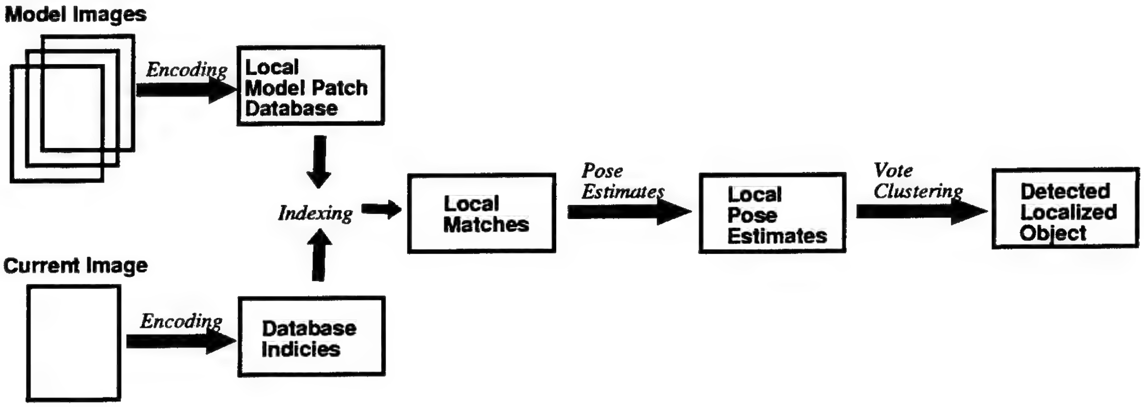


Figure 9: Basic design of recognition system that combines voting for object detection and moments for local feature indexing.

The basic approach to using brightness moments for recognition has the following steps as illustrated in Figure 9. Prior to recognition, images of the object are taken from a sampling of 3-D views. For each of these model images, local image patches are automatically selected at various image positions and scales. Each patch is encoded by a set of Gaussian derivatives. These features are normalized with respect to certain image changes and transformed in a way to be optimum for patch discrimination. Information about each patch is stored in a model data base indexed by the features.

During recognition, patches in the input image are selected and encoded in a similar fashion. Each encoded patch is used as an index into the database to retrieve potentially matching model patches, which are then tested for quality of match. Each matching pair of input and model patches then votes for the approximate object pose consistent with their match, incrementing an accumulator cell associated with the pose. After voting, the cell with the largest vote count is selected, and if the count is above some threshold, the associated object pose is returned.

5.3.1 Local representation

A local image patch is represented by a set of derivatives of Gaussians. The measurements range in Gaussian scale and order of derivative. As discussed above, it may not be advantageous to use too large a range in scales. In addition, since our system localizes the object by consensus of multiple matches, it is not essential that any given local patch be uniquely matched—as long as the overall processing time is reduced enough by initial patch matching. Thus, only a small number of scales and derivatives may be required. For the experiments presented here, two scales separated by an octave and all the first and second derivatives in the x and y directions were used; this creates ten measurements per patch.

The measurements are normalized with respect to changes in contrast and rotation in the image using the gradient of the larger scale: the responses are divided by the gradient magnitude and, as in [33], rotated so that the larger scale gradient is parallel to the x-axis. This gives us eight remaining measurements to use as features. Since all the odd derivatives, and all the even derivatives, are dependent, the feature space is transformed using principal component analysis. For patch samples from the model images used in the experiments, the resulting eight features have a covariance matrix that is approximately the identity matrix.

Experiments here and elsewhere [[33], [34]] show that Gaussian derivatives have stability with respect to view changes, and are stable enough to be useful for discrimination purposes. In [33], good matches are demonstrated after 3-D rotations of up to 22.5 degrees. Experiments here have shown stability with respect to scale changes. After changing the image scale by as much as twenty percent, the resulting feature vector's average distance from the unscaled patch's vector is very small relative to the distances between randomly paired patches. In one experiment, a threshold was selected such that 50 percent of the distances between scaled patches and their unscaled counterparts were less than this threshold. (A sample of 20,000 patches were used.) Of a population of 9,000 randomly paired patches, only 22 had distances less than this threshold. In this case, the L^1 norm was used; similar results were obtained with the L^∞ and L^2 norms.

Thus, each local patch is represented by eight statistically independent features that show a strong ability to discriminate.

5.3.2 Overall object representation

The object is represented by a set of images, taken from a sampling of views. Because of the above-mentioned feature stability, each view angle parameter (pan and tilt) is sampled at roughly 45 degree intervals, where pan is a rotation about the object's vertical axis, and tilt is a rotation away from it. For the final experiment in this section, nine view samples were used, separated by roughly 45 degrees, covering somewhat less than a hemisphere of views (approximately a 135 degree spread).

From each image, patches of different scales and positions are selected. Each patch has a scale associated with it, the *base scale*, which is the larger of the two Gaussian scales used to represent it. By using patches of different base scales, the object can be detected as it varies in size. Since the patch features used here have shown good stability for up to 20 percent scale change (approximately a quarter octave), base scales with up to half an octave of separation can be used. In the final experiment reported here, patches of three different base scales were used to cover about an octave range in scale.

With tens or hundreds of thousands of pixels in an image, the image patch positions must also be subsampled. Regular grid sampling is not necessarily desirable since the sampling in the reference and input image are bound not to match under view change. For the experiments presented here, patches were selected based on the following criteria:

- The gradient of the larger scale must be above some threshold (contrast threshold). This allows the normalization to be stable.
- The point must be a zero-crossing of the Laplacian at the larger scale. This selects a scattering of patches at about the right density for the objects studied here (human faces), and since the zero-crossings tend to be stable as the view changes, the patch positions tend to be roughly aligned.
- A point must have patch features sufficiently different from its neighbors. (Neighbor distance threshold.)

Following these policies creates roughly 150-250 patches for a 256 by 240 image and a base scale (Gaussian σ) of 5 to 7 pixels. With three scales and nine views, roughly 4,000 to 7,000 patches are created to model the object's appearance. It is important to note that the zero-crossing restriction reduces the number of patch features to seven, since the two second derivatives at the larger scale are now dependent. However, they are effectively used to discriminate the selected patches from the great majority that are not selected.

5.3.3 Indexing

Since indexing is not the final step of detection in our design, and is only used to make the matching at the local patch level efficient, complex indexing strategies such as in [34] have not been stressed. (Though they may enhance the performance.) Instead, the features of a patch are quantized, and the quantized vector is used as an index. To allow for some additional feature variation, the quantization ranges overlap by some amount (quantization overlap). This means that there will be multiple entries in the table per model patch stored, but, during recognition, the input image patch indexes only a single cell, and thus, is very efficient.

For the final experiment presented here, a quantization level of three buckets was used. This gives us a table of $3^7 = 2187$ distinct cells. The overlap policy produces about 10 entries per stored patch, or roughly 40,000 to 60,000 entries. The retrieval rate (the number of model patches retrieved per input image patch during matching) ranges from 30 to 70. Considering that there are 4,000 to 7,000 patches used to represent the 3-D object in a wide range of poses, this implies that, at most, one percent of the model patches are selected during indexing.

The model patches retrieved by an input patch index are then matched against the input patch. As discussed above, different norms have been compared for discrimination effectiveness under image distortion and have been found to be similar, with the L^1 and L^2 norms performing best. The L^2 norm was used for the recognition experiments discussed below, though the L^1 may be more efficient on certain computers.

5.3.4 Voting and detection

Once a model and input patch match, information about their positions, scales and orientations of their gradients, as well as information about the 3-D view associated with the model patch, can be used to roughly estimate the object pose transformation consistent with match.

Stored with the model patch is its scale and its 3-D view in pan and tilt parameters; this allows a rough estimate of these pose parameters. The 2-D rotation between the patches can be estimated by the difference in the gradient orientation of the two matched patches. This leaves the parameters representing the object translation in the image plane.

Image translation is estimated in two steps. During model image capture, a point on the surface of the object is tracked from frame to frame and is used as an arbitrary origin for the object (Figure 10). For each model patch, the image position of the origin relative to the model patch is stored with the patch. This relative position is in a reference frame defined by the position of the model patch in the image, its base scale and the orientation of the gradient at the base scale.

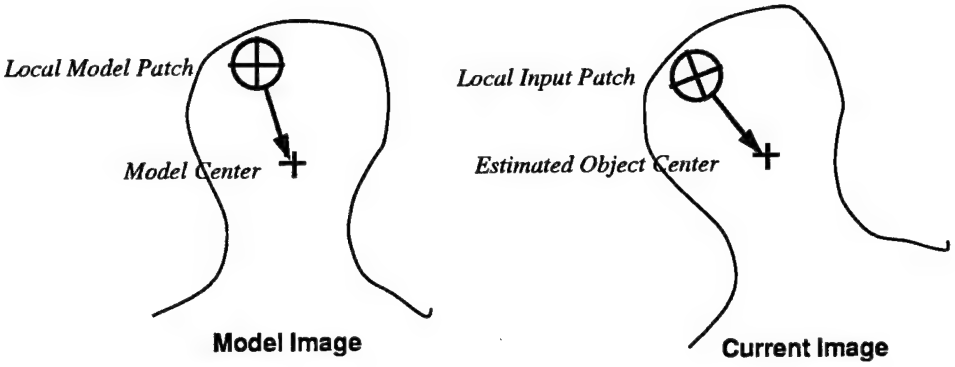


Figure 10: Pose and position voting. Given matching model and input patches, a vote for the position of the object center in the current image is generated. It is estimated from the stored position of the center relative to the model patch and the computed transformation between the local reference frame of the model match and that of the input patch.

During recognition, the object origin will have approximately the same relative position with respect to the correctly matched input patch as the model patch, and the origin's absolute image position can be recovered given the position, gradient orientation and scale of the matching input patch. This object position calculation during recognition is very simple, making the overall pose voting step very efficient. The pose estimate is also fairly accurate, producing clear clusters in pose space when sufficient matches are generated. Figure 11 shows an example of this. The accumulator is shown as an image with brightness representing vote count, and the horizontal and vertical axes represent image x and y respectively. The bright spot is a concentration of votes at the correct pose.

For the final experiment presented here, the 6-D voting space included the full range of positions possible in a 256 by 240 image, 180 degrees of range for each rotation parameter (pan, tilt and rotation in image) and an octave range of scale. The voting accumulator was quantized so that x, y, image rotation, scale and tilt had 32, 30, 4, 3 and 3 cells respectively. Pan was given only one cell (different pans were not distinguished), to keep the overall size of the accumulator down to an efficient size.

5.3.5 Experiments with brightness moments

In the first experiment, a single model image is compared to a range of input images to demonstrate the stability of detection. Figure 12 shows sample images from a twelve frame sequence with changing features (eyes closing and mouth opening) and view. The first ten images were strongly matched—the correct peak in the accumulator was four times higher than any random cluster, where a peak is judged correct if it is within

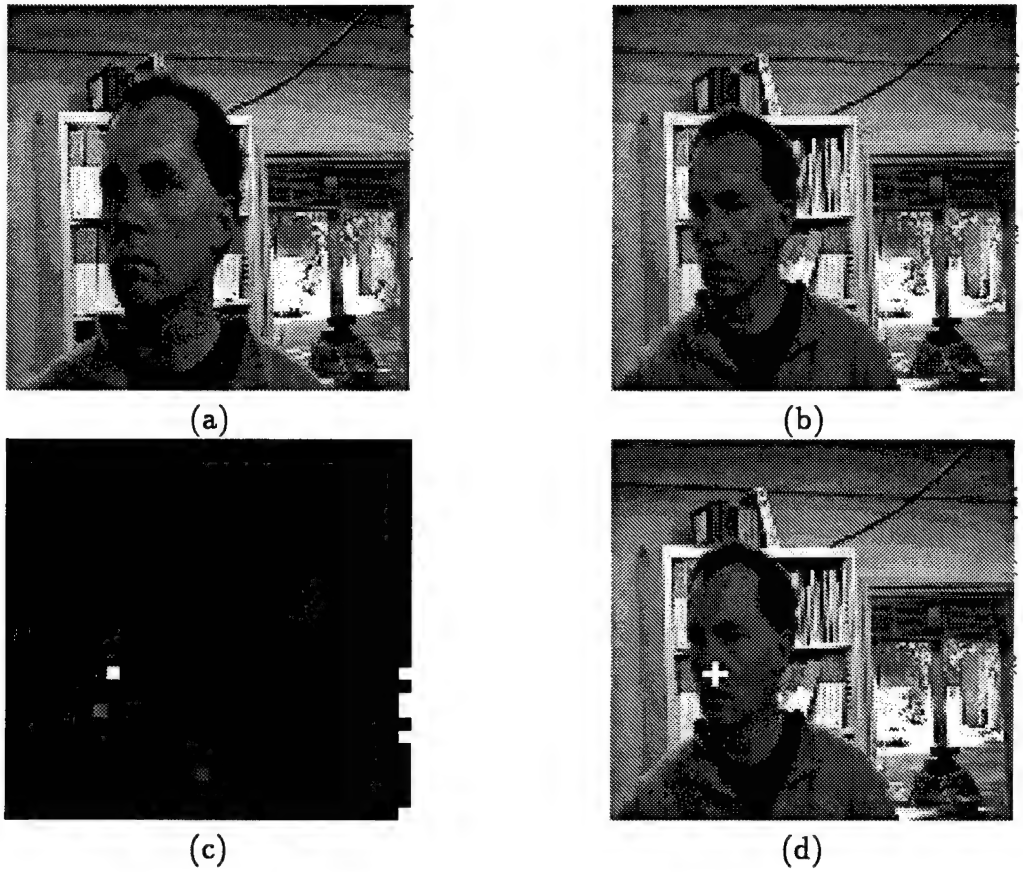


Figure 11: Detecting clusters of votes in pose space. (a) Model image. (b) Current input image. (c) Slice of pose accumulator showing vote distribution across cells representing image positions x and y (horizontal and vertical axes respectively.) Brightness of cell represents vote count, with cell of correct pose forming a clear, strong peak. (d) Resulting detected and localized object. (The subject's nose is arbitrarily used as the object origin in all experiments presented here.)

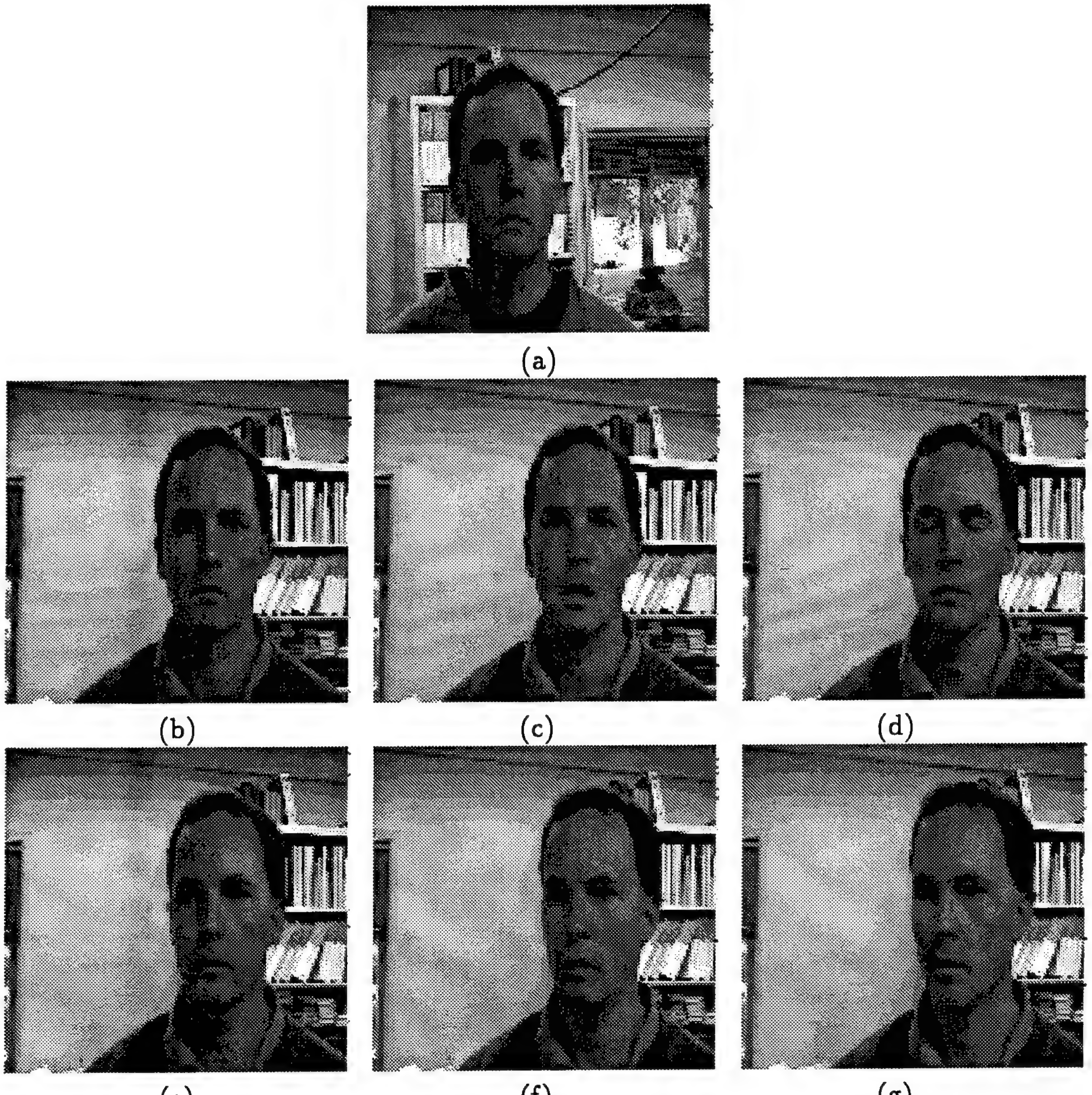


Figure 12: Experiment demonstrating stability of detection using one model image. (a) Model image. (b-e) Examples of the 10 out of 12 frames that had clear, strong and correct peaks, in spite of feature movements and head rotation. (f) Frame with correct, but weaker peak. (g) Frame with over 40 degrees of rotation, some scale change and failed detection.

approximately one accumulator cell of object origin's position. These ten strong matches include those to the first four images shown in the figure. In the eleventh image, the correct peak is still the highest, but is only slightly higher than the clutter. In the final image, the rotation is beyond the range of the model image and the correct peak is not selected by the matcher. However, this rotation is over 40 degrees and the scale has also changed. Overall, this sequence shows the potential usefulness of the method. In these tests, the best performance was achieved when the contrast threshold was set to 5.0, the minimum neighbor distance set to 0.5, and the feature quantization overlap was set to 0.3.

In the second experiment, a person is detected in a 150 frame sequence of him talking, blinking and rotating. Three model images were used, shown in Figure 8; the four images shown in the figure display the general range of variation. The rotations included rotations in the image plane and about a vertical axis (pan) for a range of over ninety degrees. The model images were selected at three different pan positions, and the system correctly detected and localized the face in each of the 150 frames. The correct peaks were several times higher than the random clutter.

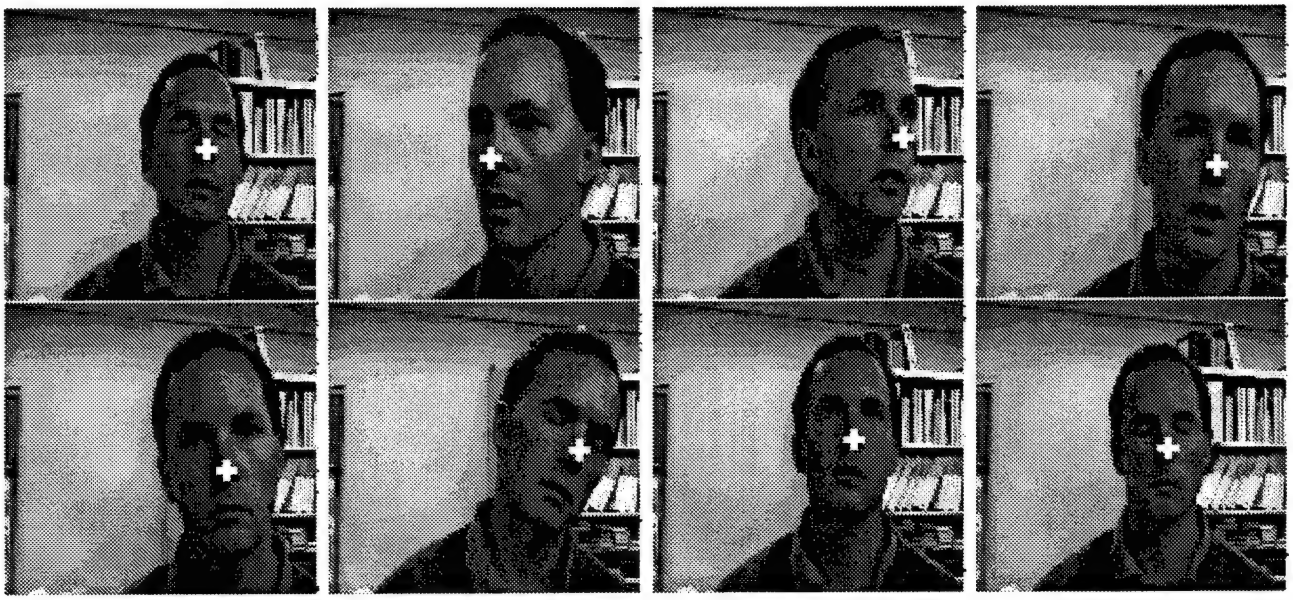


Figure 13: Experiment demonstrating the full system with multiple model images and a 100 frame sequence. For 98 out of 100 frames, the cell associated with the correct pose was the accumulator peak, and was almost always clear and strong. The eight correctly matched samples shown above demonstrate the range of pose and feature changes.

In a final brightness moment experiment, the full 6-D pose system was tested with all the parameters set as discussed above (Figure 13.). The subject was allowed large motions in all six degrees of freedom, as well as talking, blinking and other feature changes; the scale changes covered over half an octave (50 percent change in scale). One hundred frames were grabbed over a 20 second interval. In 98 out of 100 of the frames, the system correctly detected and localized the face, and, for an overwhelming majority, the correct peak was at least two to three times higher than any random clustering of clutter in the accumulator. For the two frames with bad matches, one had a clear, strong peak near the object origin, but not within one accumulator cell (it was more than four cells away). This may have been caused by the fact that the actual object 3-D orientation was between those sampled for modeling, producing errors in the origin estimate. In the other bad match, there was a clear, correct peak, but it was not high enough above accumulated clutter in other parts of the voting space.

5.4 Lighting change and orientation fields

For recognition to be successful, the detection rates must be stable with respect to large changes in lighting. The above brightness moment system is invariant to changes in image contrast through normalization of the features; however, it is not explicitly tolerant of large changes in the direction and distribution of light sources. Figure 14 shows the effect of such changes on the appearance of an object. Much of the brightness variation in the image of a face is in fact caused by shading. By changing the lighting, the shading and hence the brightness moments representing it, undergo large changes. These changes can often include complete reversals in sign.

Is it important to work with properties of the image that tend to be stable with respect to lighting changes. One property that is often stable is the general direction of the brightness variation modulo 180 degrees: even though the magnitude may vary and the sign may flip, the direction of the gradient is often constrained to lie near a line. Figure 14 shows the orientation fields for two different lighting conditions. This stability is certainly true at the projection of many types of edges, including physical, occluding and reflectance. In addition, this tends to be true in shading. Over a significant range of light source directions and object surface curvatures, the gradient orientation lies near a line parallel to the projection of the direction of maximum magnitude curvature of the surface being illuminated. The more extreme the ratio of principal surface curvatures the more this is the case: for cylinders, the shading gradient is almost always so oriented. For many surfaces with more finite ratios, this is still true over a large range of light source positions.

For this reason, we developed and explored a method of using moments of the gradient orientation field (modulo 180).

5.4.1 Local representation

The local representation requires that we compute the average gradient direction (modulo 180) within a Gaussian weighted window of variable size. One measure that has this property is the eigenvector associated with the minimum eigenvalue of the smoothed

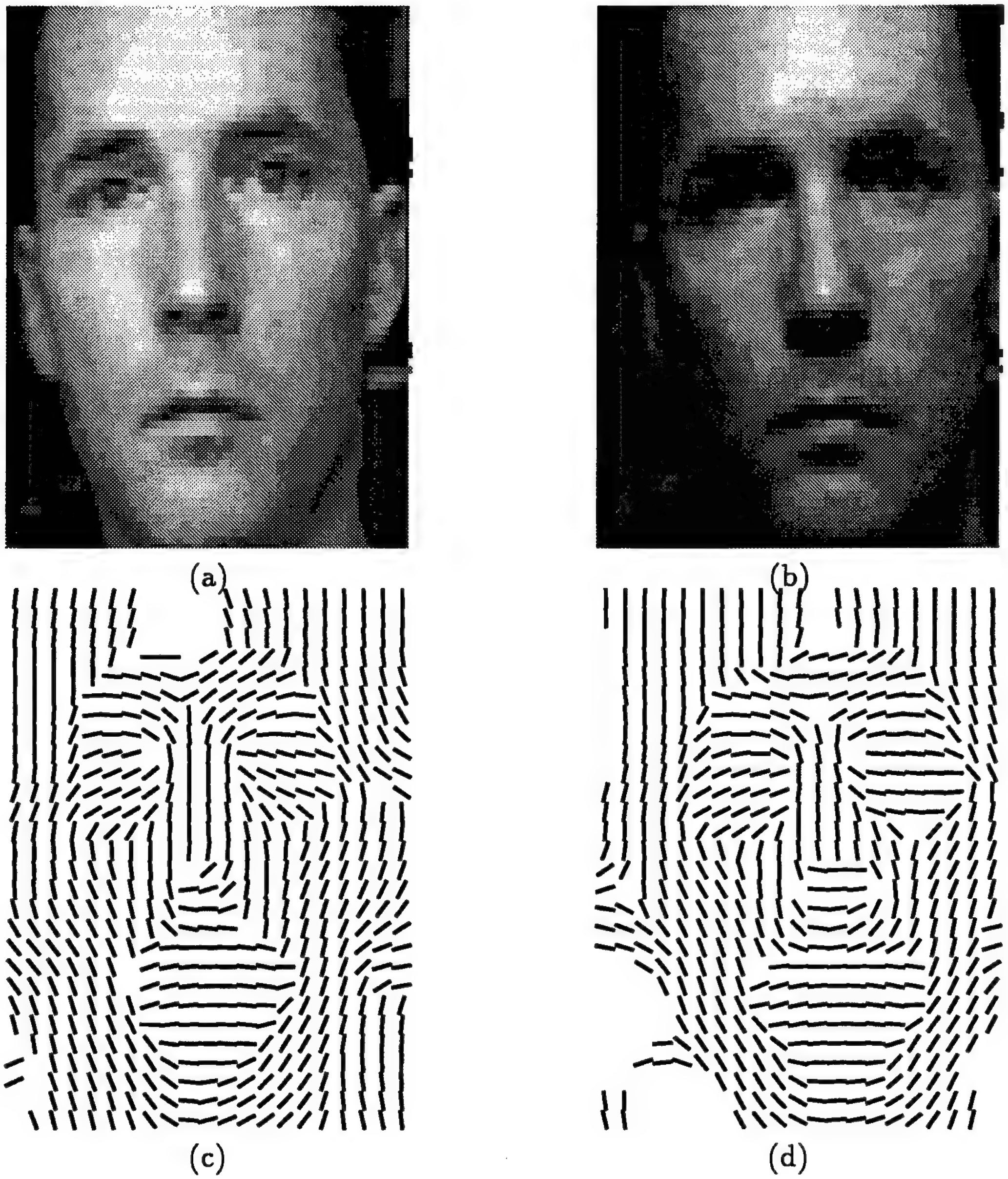


Figure 14: Example of object appearance under large lighting changes. (a - b) Two images of a face with different lighting. (c - d) The local texture orientation (normal to the gradient and modulo 180 degrees) demonstrating the potential stability of this local feature.

texture matrix of Lindeburg and Garding [39]:

$$W \left| \begin{matrix} g_x^2 & g_x g_y \\ g_x g_y & g_y^2 \end{matrix} \right|$$

where $(g_x, g_y)^T$ is the gradient of the image convolved with the Gaussian at the *texture scale*, and W is the Gaussian weighted averaging of the matrix terms at the *integration scale*. The resulting eigenvector reflects the texture orientation θ in the underlying image, and is invariant to the sign and magnitude of the texture contrast. Figure 14 shows the resulting θ fields, where the bar directions are actually aligned with the texture orientation (normal to the gradient orientation).

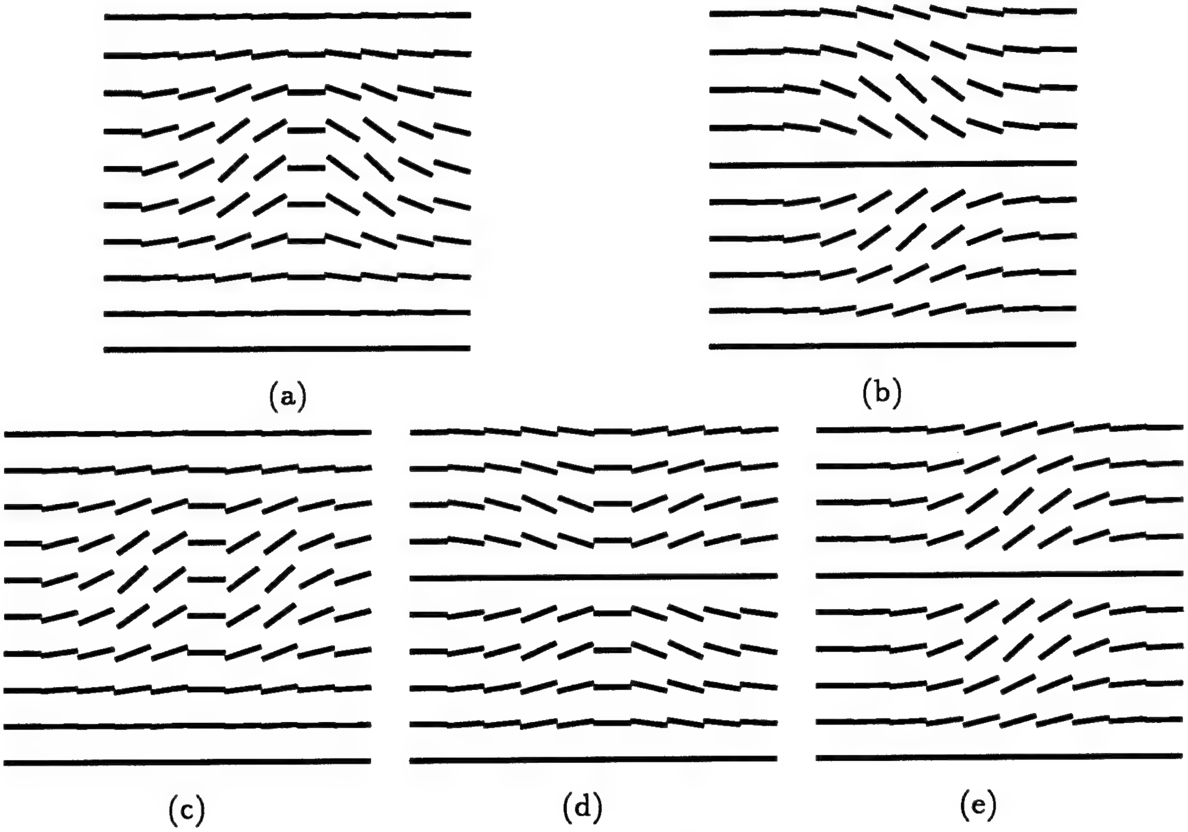

(a)
(b)
(c)
(d)
(e)

Figure 15: Examples of orientation fields that the different first and second derivatives respond selectively to. Each of the five fields produces a significant response from one of the derivatives and zero from the others: (a) θ_x, (b) θ_y, (c) θ_{xx}, (d) θ_{xy} and (e) θ_{yy} respectively.

By choosing different integration scales and differentiating the resulting orientation fields with respect to the image x and y, we have multiscale orientation moments. Figure 15 gives a feel for the different characteristics of the fields that the different first and second derivatives are sensitive to. Each of the five fields produces a significant response from one of the derivatives and zero from the others: (a) θ_x, (b) θ_y, (c) θ_{xx}, (d) θ_{xy} and (e) θ_{yy} respectively.

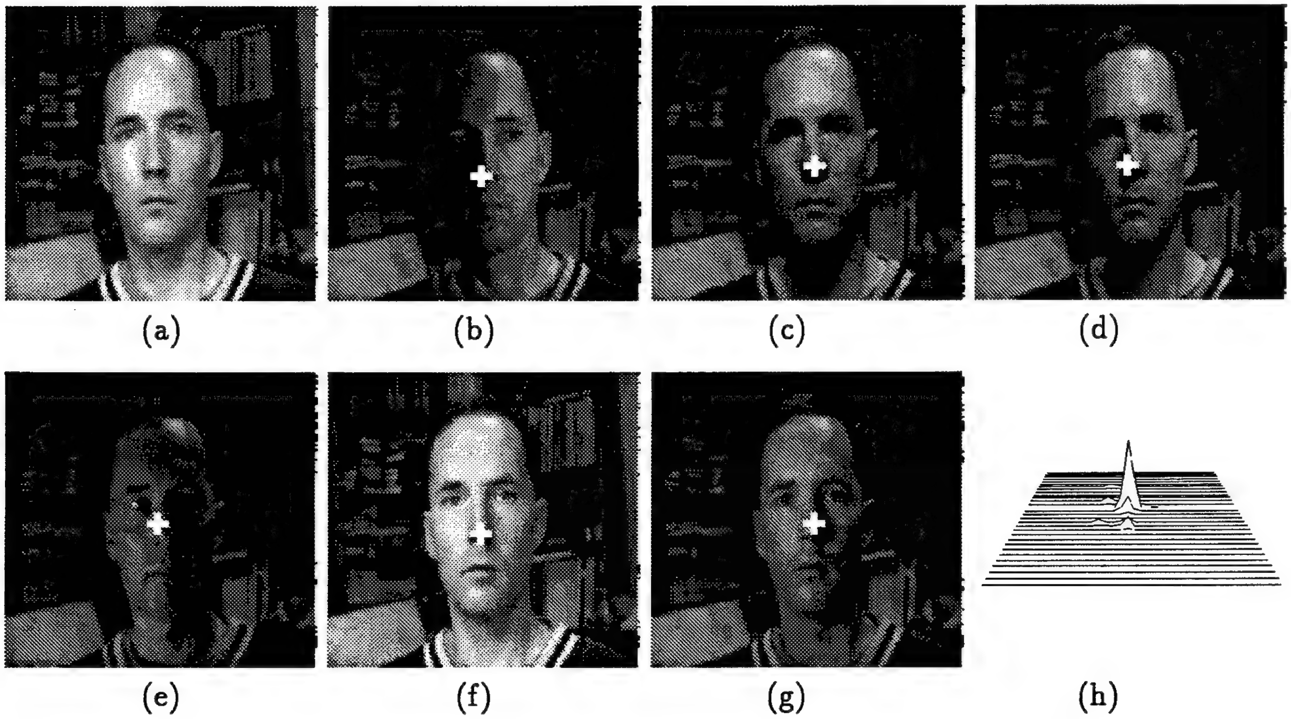


Figure 16: Demonstration of recognition based on moments of orientation. (a) Model image. (b-g) Six of the 25 images taken under varying lighting; 24 of the 25 were correctly matched, with strong peaks, including above six. (h) The (x, y) projection of vote accumulator of image (g), showing clear peak.

For our experiments, we used two integration scales, an octave apart, and all first and second derivatives of the orientation at each scale. The values are normalized with respect to image rotation by rotating (steering) the direction of differentiation to be parallel x' and orthogonal y' to the orientation field at that point. Principal component analysis of the data is also done, as in the brightness data. This gives us ten normalized and orthogonal features for local patch matching.

5.4.2 Experiments in lighting tolerance

The detection system using orientation moments is essentially the same as the brightness system discussed above. The major difference is that the quantization level was set to two: only the sign bits of the features were used. It was unclear how stable these feature are, and, since there are three more features per patch, we could afford to use less information per feature. Another difference is that the sign ambiguity of θ (modulo 180 degrees) creates an ambiguous pose estimate: two poses are consistent which each patch match, hence two votes are generated.

Figure 16 shows the results from an experiment where the lighting was varied dramatically and everything else was held roughly constant. Twenty-five images were taken, with the light source ranging in the pan and tilt directions by more than 90 degrees. (The light was approximately one meter away.) In some frames, this concentrated light source was the sole source, while in others, a large, strong diffuse source was added (a large window). One image was selected as the model (Figure 16), and all were matched to it. In twenty-four out of twenty-five of the images (96 percent), the correct peak in the voting space was selected. It also was typically strong and clear (Figure 16). In the miss-detected image, the correct peak is still clearly discernable, but some spurious clutter created a higher peak elsewhere.

5.5 Summary of consensus and moment based results

This study combined two useful methods in recognition: consensus or voting-based approaches and moment-based representations. This combined method is an improvement over voting and moment methods in isolation. Using image brightness moments, the idea is successfully demonstrated on examples of human faces undergoing full 3-D pose change, as well as changes in features such as talking and blinking. The idea is then extended to moments of local texture orientation and successfully demonstrated under large variations in lighting.

Overall, the detection rates are very good for large ranges of 3-D poses. The system also has the potential to be fast during recognition. Gaussian convolution can be very fast and only two scales are used. The finite differences and feature transformation required are simple and need only be performed on the roughly 1,000 patches selected per input image. In addition, for each input patch, the indexing is very fast, and with an average of 30-70 model patchs retrieved, only 30,000 to 70,000 patch comparisons are performed. Future work includes a real-time implementation on a conventional computer. Achieving real-time recognition is part of the motivation for the design.

One area for continuing improvement is the height of the correct peak relative to the height of spurious peaks generated by random clutter. One method of doing this is to increase the number of features used to filter out more bad patch matches—perhaps by using the first three Gaussian derivatives. Another valuable experiment is to combine the use of both brightness and orientation moments. The former represents information that many times can be useful (e.g., the sign of contrast), while the latter should be more robust in many other situations.

6 Enhancing consensus and moment-based recognition

In this section, we report on enhancements of the recognition system discussed in the last section. The goal in this work is to achieve even higher reliability in the matching process as the head of the tracked subject undergoes significant motions and changes relative to a given model image.

In the recognition studies described here, a single frame was used as a model image of the head. In practice, the recognition system will generally have more stored model images of the sought-for subject. However, there may often be ranges of views for which the subject is not modeled, thus, it is important to demonstrate the ability to match over a range of views with a given model image. In the important case of police photo files, there may only be full-face and profile views of a person; thus, a security system may have to recognize a face from a view that differs from the model views by as much as 45 degrees.

In addition, the work in this section goes beyond the previous work in that it applies the matching output to the task of discriminating between multiple subjects undergoing motion. The discrimination experiments involved over 300 images and demonstrated significant improvements over the previous system.

The recognition system reported here follows three steps: (1) measure image features, (2) rapidly detect rough matches between image and modeled object, and (3) refine and evaluate match via object localization.

In the first step, Gaussian-weighted moments are again used, but are combined with semi-local geometric features measured over neighboring sets of detected points in the image. This idea adds more features and provides for a stabler moment normalization.

In the second step, the matching is determined by analyzing semi-local match consistency. This method can potentially find the correct match under greater deformations of the model image than the rigid pose-based method described in the last section. Experiments indicate generally more competent matching of the new method when the object is undergoing large 3-D rotations. Since most of the processing in the new method is in simple indexing and voting steps, efficient implementation should be possible on conventional computers.

The last step in the process introduces a global analysis of the model-to-data transformation that tolerates large errors in the local matches.

The recognition system is described and motivated in more detail below. This is followed by a demonstration of its usefulness in discrimination tasks.

6.1 Features

The features used to recognize objects should be (1) simple, generic functions of the image data, (2) reasonably stable with respect to global transformations such as 3-D rotation of the object, (3) capable of representing independent pieces of information at many different locations distributed about the image of the object, and (4) fast to compute.

These properties can be satisfied by using Gaussian-weighted image moments, especially when they are suitably normalized, sampled at Laplacian of Gaussian peaks, and augmented by semi-local geometric features.

Image moments are an infinite series of terms that describe the distribution of some type of value in the image [32]. The example explored in this work is image intensity, though other types of image data, such as color or texture, can be similarly represented. The different terms in the series characterize different aspects of the intensity pattern in the image, and they tend to be very efficient in their representation: the first few terms characterize much of the overall pattern—they are simple and direct functions of the data, meeting the first requirement. Also, since the lower terms represent coarse features of the pattern, they are relatively stable with respect to transformations of the images. In fact, they can be explicitly normalized with respect to changes in brightness, contrast, rotation in the image and other transformations. Thus, using the first few image moments, normalized, helps to satisfy the second requirement.

Gaussian-weighted moments have the additional advantages of locality and efficient implementation. By modifying the mean of the Gaussian weighting function, the position of the data contributing to the moments shifts in the image, and by modifying the Gaussian scale factor (σ), the size of the contributing region also changes. Thus, the position and size of the image locality being measured can be varied. By measuring the moments over a range of positions and scales, the third requirement is satisfied: a large set of spatially distributed features are available for recognition. Along with being localized, Gaussian-weighted moments help satisfy the final requirement: they can be efficiently implemented. They are simple, linear combinations of the derivatives of the Gaussian filter, which in turn, is separable and be rapidly computed. In fact, the real-time detection and tracking module discussed in the previous section already computes two of the second derivatives everywhere in the image in a few milliseconds on a 66 MHz Pentium.

The moments computed at two close positions and scales are highly correlated. To compute a minimal set of measurements that are reasonably independent, the moments should be applied at selected positions and scales. The peaks of the Laplacian of the Gaussian (LoG) of the image are reasonably stable as the image changes, and if the scale of the peak detection is matched to the scales used to compute the moments, the spatial sampling induced by the peak locations is a good compromise between maximizing the number of feature measurements and their independence. For the scales used in the current study, restricting the moment measurements to the peaks reduces the number of moment measurements from tens of thousands to a few hundred. This increases efficiency dramatically and still provides hundreds of features for recognition.

Thus, Gaussian-weighted moments measured at LoG peaks helps satisfy all of the above feature requirements. The usefulness of moments can be further enhanced by semi-local analysis. This is a combination of localized moment measurements and a geometric analysis of multiple measurement points (peaks) over a neighborhood of the image. The geometric arrangement of the neighboring peaks can be used both to normalize the moments measured at that peak and to add new (semi-local) features.

For the moment measurements to be stable with respect to image transformations, it is important to normalize them: to factor out the effect of the transformations. For example, orienting the moment measurements in the direction of the local gradient (the first moments) tends to cancel-out the effect of rotations in the image. However, since the local gradient can often have a low magnitude response, it is not always possible to reliably use it to normalize all the moments at a given point. In fact, all of the derivatives (moments) have this problem. However, the angular position of a neighboring LoG peak relative to a given peak can often be stable and is used here to normalize the other features with respect to rotation. The relative distance of the neighbor to the given peak is also a measurement that can be exploited. In the current design, the angular position is used to normalize for rotation, and the distance as an additional feature to match. By using only immediate neighbors of a given peak and employing appropriate data structures for search efficiency, this semi-local analysis can be made fast and enhances the usefulness of moments.

In the current prototype, the first five moments (a row in Figure 8) where measured at every LoG peak, and the closest ten or so neighbors were used for semi-local features. For each pairing of a peak with one of its neighbors, the moments of both where normalized by the angular position of the neighbor, and their distances and brightness differences were used as additional features. This gives twelve semi-local features for every peak-neighbor pair. Since the measurements are occurring at peaks of the LoG, the LoG is naturally strong at these points. Thus, the LoG magnitude is used to normalize the moments and the brightness difference with respect to changes in contrast.

Finally, since moments are measured only at LoG peaks, the second moments are correlated. This requires us to use their sum (LoG) and difference, rather than the second moments directly. Since the magnitude of everything is normalized using the LoG, this feature only contains sign information. However, since LoG peaks of both signs are being selected and the magnitude tends to be strong at peaks, the sign information is very stable and useful.

The above allows us to describe every LoG peak in terms of a set of twelve-element feature vectors, one vector from each of the ten or so neighbors. Collectively, this set of features provides a fairly unique, stable and easy-to-compute representation of the image region about the peak. The reliability of the identification is furthered when these measurements are made at many peaks distributed about the image of the object.

6.2 Approximate match analysis

Given the above features computed over the image, or in regions of interest, the next step is to find likely matches between this data and similar, pre-stored features of the sought-for object. The present matching algorithm does not specifically require peaks of the LoG, thus, the peaks of the previous subsection will be more generically referred to as points.

There may be many points (peaks) in both the model and current region of interest, therefore, searching through the set of possible permutations of point correspondences for the best match may be prohibitive. What is required is a match analysis that is fast, but sufficient to detect a strong match of the object when it is present and little when it is not. Thus, an efficient method of computing an approximation of the best match is desired. In addition, the method must accommodate distortion: it cannot just rigidly translate the model, searching for the best model-to-image point alignment. The latter requirement is referred to here as flexible matching.

The simultaneous requirements of speed, sufficient evidence of match and flexible matching, imply a matching process that starts with a very rough, many-to-many point match, and incrementally refines this match until the overall quality of the match can be used to determine if identification has been achieved. It is possible to efficiently design the computation of both the rough match and the incremental refinement, and the incremental refinement need only cycle until the overall confidence in the match is either sufficiently high (accept) or low (reject).

The rough matching step can be effectively implemented using efficient indexing and voting methods, and the refinement step can be effectively implemented using local match consistency filtering, followed by outlier analysis during the localization step. The last operation will be discussed in the next section.

More specifically, in the design presented and demonstrated here, the rough matching step proceeds as follows. Each detected point is associated with approximately ten neighboring detected points, and the twelve semi-local features defined in the last section are computed for each point-neighbor pair. For every detected point in the image region of interest, and for every one of its neighbors, use the twelve features to retrieve matching point-neighbor pairs of the model image that are stored in a data-base indexed by the same twelve features. For every model point-neighbor pair retrieved by the current image point-neighbor pair, vote for the match between the associated current image point and model image point. Point matches between the current image and the model image that receive multiple votes—multiple neighbors with similar relative properties—are much more likely to be correct than those that do not. Thus, the overall, initial rough match between the images is defined to be the set of point-to-point matches with vote counts above some threshold.

In the current implementation, the data-base of point-neighbor pairs is indexed by a binary number, where each bit of the number represents a binary quantization of a separate point-neighbor feature. For every model point-neighbor pair retrieved, its unquantized feature vector is correlated to the unquantized feature vector of the current image pair used to retrieve it. The correlations are normalized by the magnitude of the features, and correlations above 0.78 are accepted as voting point-neighbor pair matches. If any match between a model point and a current image point receives 4 or more votes, that match is added to the overall rough image match.

For the image pairs tested there were typically 300 points per current image and 60 points per object model region. This gives a typical total of 18,000 possible point matches. The rough initial matching process typically produced a total of 100 point matches, or a reduction factor of 180. Since efficient implementations are possible for the feature detection, indexing and voting steps, the overall rough matching process should also be fast.

Even though this is a large reduction in point-to-point matches from the total possible, most of the matches are still erroneous. Since the number of point matches are used as a means of classifying the image, the initial point match set must be refined till most of the matches are likely to be correct. It is important to note that this does not mean that the resulting set of point matches has to be perfectly correct, just representative enough of the targeted match. For this reason, a refinement process is employed that is simple and fast, but may stop short of guaranteeing a completely correct point match.

The match selection criterion for each iteration of the refinement is analogous to the initial rough matching. The only difference is that the neighbor matches used in the voting step are required to have been previously accepted in the last iteration. Since they were accepted, because they, in turn, are consistent with some minimal set of other neighbor matches, this iterative refinement tends to enforce approximate consistency over large sets of point matches and remove point matches not consistent with these large sets. Since only a small percentage of the total possible point matches are considered during the refinement step, and the selection criterion involves a small number of neighbors, each refinement cycle is very fast.

Typically, it was found that the refinement process does a very good job of throwing out erroneous point matches. The total number of point matches typically starts at 100, and, after 2 or 3 iterations, it is reduced to approximately 15 to 20, with almost all being correct matches. This is true even for images of objects undergoing 3-D rotations of up to 45 degrees. For the design and demonstration discussed here, refinement is performed for 5 iterations.

6.3 Localization

Once a reasonable correspondence between model points and image points is produced, it can be evaluated by analyzing the fit of a parametrized image transformation to the point mapping. Given a sufficiently sophisticated parametrization and a method of detecting and removing inconsistent point matches (outliers), the number of remaining point matches can be used as a criterion for accepting the overall match.

Since the match procedure discussed above is restricted to semi-local match consistency, a global transformation fit analysis can often further reduce the chance of false identification. However, without the initial match analysis, the point match set would contain too many spurious matches to make localization with outlier analysis feasible.

For human heads undergoing changes in position and orientation relative to the camera, a six-parameter affine transformation is a reasonable, rough approximation of the point mapping from a model image to the given current image. This has been shown in frame-to-frame tracking using a fixed reference image and a range of head motions [40]. Given the experimental results presented below, it seems sufficient for matching methods developed for re-acquisition and identification tasks.

The initial point matching step seems to output a low enough percentage of spurious point matches to support a relatively fast and straight-forward outlier analysis during transformation fitting. This especially will be true when the recognition module is operating in conjunction with the detection and tracking modules. These latter modules provide a rough segmentation of the image, removing distant parts of the image from consideration in the global match analysis. Given this combined initial filtering of point matches, it is feasible to employ M-Estimation as a method of determining point match consistency with a global transformation. Thus, a version of M-Estimation for affine transformation fitting was implemented that is analogous to that used in [41] for 3-D object pose analysis.

The number of point matches remaining after the analysis was then used to decide if the object was present in the image. The threshold for this number was determined empirically and depended on the total number of correct matches possible for a given object model image. The specific thresholds used are discussed in the feasibility analysis below.

6.4 Scientific contributions of this recognition method

Indexing and voting methods for detecting matches have been described in [28], [29], [42], and others. These methods have been applied to detecting rigid, well-modeled 3-D structures in images using well-localized physical structures, such as straight edges and elliptical arcs. In the current work, indexing and voting is applied as a first step in a process capable of flexibly matching objects with poorly known 3-D structure directly to simple functions of the image data. The indexing and voting is done with respect to a strictly local and geometrically loose reference frame about the detected points. The detection of a global match transformation is not done during indexing; it is done afterwards, using a large fit-error tolerance. Thus, an approach to matching flexible, poorly modeled objects is contributed, and has the potential speed of traditional indexing methods.

In [34] and [35], methods of representation based on Gaussian-weighted moments also are presented. However, in both of these schemes and others utilizing moments, the moments of the object model image are measured only at a single, or a few, positions. The typical system depends on the measurement of a large, complex series of moments (up to 45) at these few, select positions for a sufficient set of discrimination features. This makes the match response very sensitive to occlusions, image clutter, and other disruptions in the data at those few points. A spatially distributed application of moments was developed here, followed by a process of using the features at all the measurement points in a global match analysis. In addition, a method is contributed that uses semi-local measurements (relative positions of points) to normalize moment measurements, and augments them with additional features.

6.5 Demonstration of feasibility

The two main requirements for feasibility are that the recognition success rate is good enough in the application operating conditions, and, a real-time implementation is feasible. Given the domain of security and surveillance, an appropriate test of feasibility is the matching and discrimination of human heads undergoing normal motions.

This section has three parts: The first is a demonstration of the matching process in this context; the second is a demonstration of the use of the match response to discriminate multiple, similar objects; and the final part is a general discussion of the feasibility of satisfying the recognition requirements.

6.5.1 Matching under appearance variation

In security and surveillance situations, the head of a tracked subject can be expected to rotate, change in scale and position, and undergo deformations associated with talking and facial gestures. In the recognition studies described here, a single frame was used as a model image of the head. In practice, the recognition system will generally have more stored model images of the sought-for subject. However, there often may be ranges of views for which the subject is not modeled, thus, it is important to demonstrate the

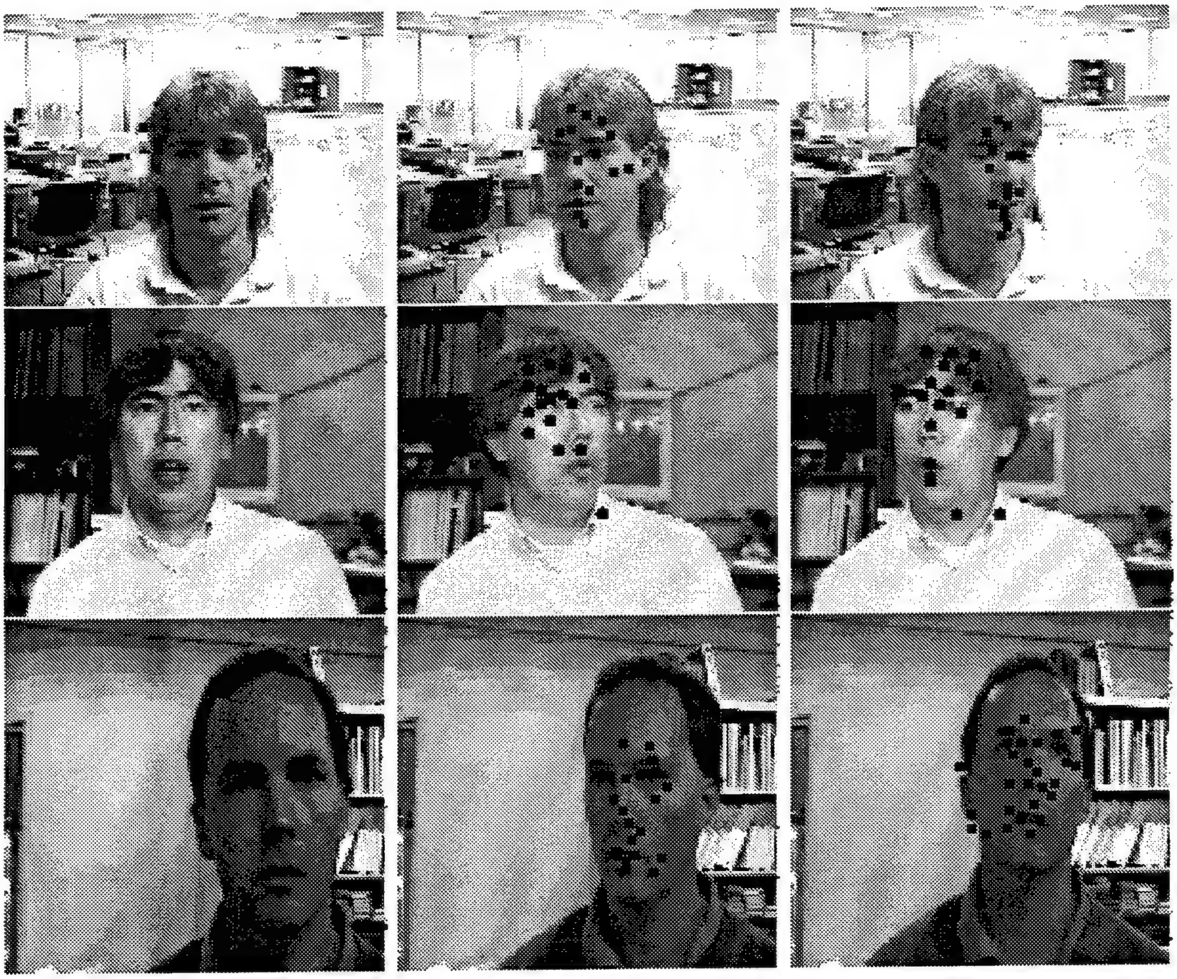


Figure 17: Examples of point matches. First column: model image. Other columns: matches between respective model image and other views of the object. Dots are LoG peaks that have been matched to model image peaks.

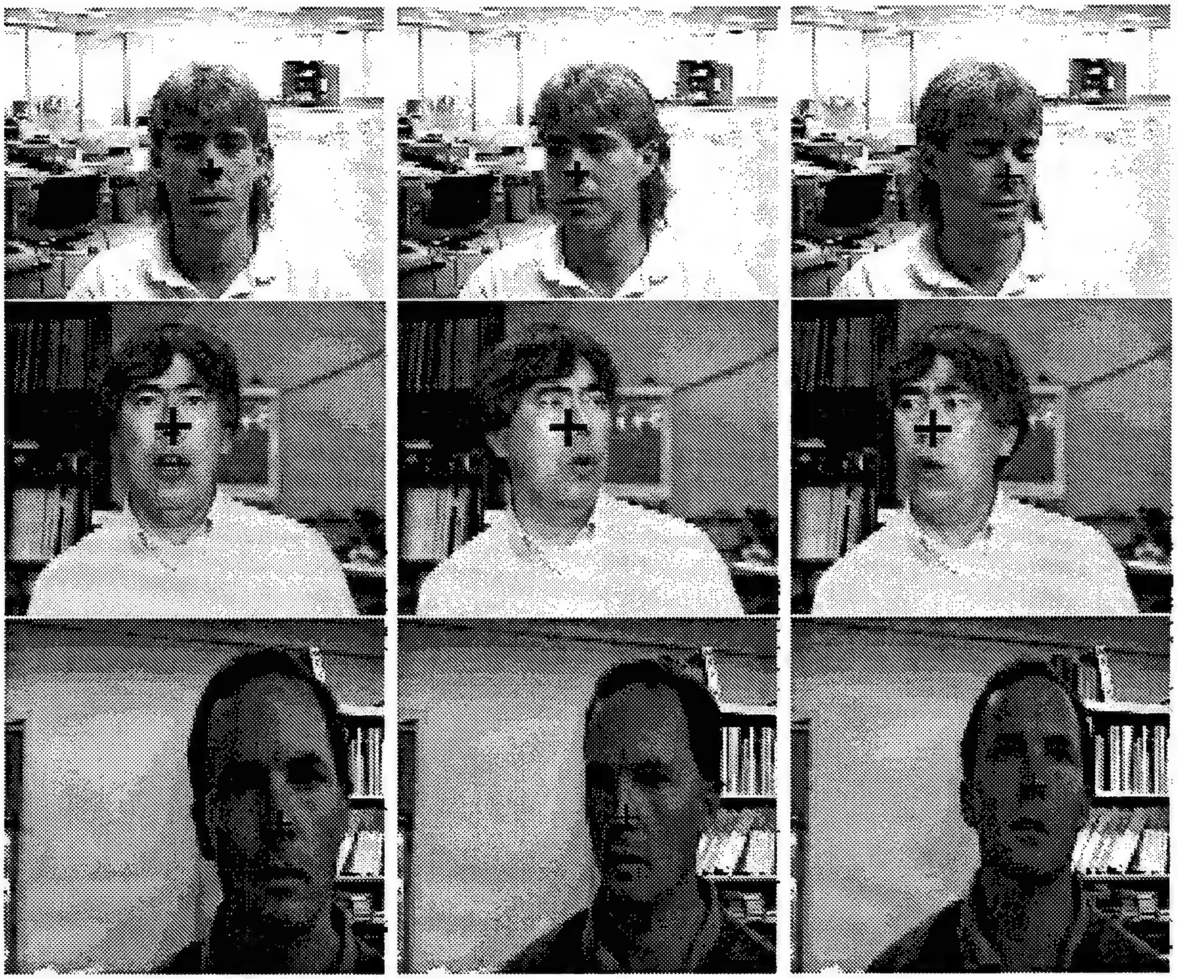


Figure 18: Examples of match-estimated object location. First column: model image showing nose location. Other columns: crosshair shows match-estimated location of model nose in other views of the object.

ability to match over a range of views. In the important case of police photo files, there may only be full-face and profile views of a person, thus, a security system may have to recognize a face from a view that differs from the model views by as much as 45 degrees.

The matching output was considered useful if the great majority of matching peaks in the current image were in the head region and the estimated transformation given the matched peaks seemed reasonable. A quantitative study of discrimination rates based on this matching output is discussed in the next subsection.

The video data included rotations of the head of over forty-five degrees from the model view, and changes in object scale of over 24 percent. No limits were placed on where in the image the head could be, or how it was oriented in the image (rotation in the image plane). In fact, the performance of the system would have been the same if the head had been upside down. Three video sequences were studied, each containing a different human subject undergoing motions. Two of the sequences covered over 20 seconds of continuous motion, and a total of 350 frames were used. In each sequence a frontal view was selected as the model image, and, then was matched to all frames in the sequence. The recognition process was applied to image data using a Gaussian scale (σ) of 3.75, and all of the other system parameters were set to the values described above.

For almost all of the frames, the matched points were overwhelmingly concentrated in the head region, and the estimated match transformation seemed reasonable. The latter was observed by using a crosshair to indicate the nose position predicted by the transformation. In almost every case, the crosshair was on or near the nose. Figure 17 shows examples of the matches. The first column shows the model image, and the other columns show matches between the model image and typical views of the object. The dots indicate LoG peaks in the current image that have been matched to the model image. In Figure 18, the crosshairs indicate the match-estimated location of the model image nose in the current image. The matching and localization of the objects seem reasonable in spite of the large changes in view and variations in facial features.

This demonstrates that the matching system is capable of tolerating view angle changes of in a practical range. In addition, it handles changes in scale of up to 24 percent. As discussed in previous sections, the detection and tracking modules are capable of isolating regions of interest in every frame. The extent of these regions can be used as rough approximations of the scale of the object to be identified. Given these rough approximations of scale, the scale variation tolerance of 24 percent exhibited by the system demonstrates that practical performance levels are feasible here also.

6.5.2 Discrimination of similar objects

This section describes the quantitative study of the recognition system applied to discrimination of the similar objects. The objects were the three human heads shown in the previous section. Human heads tend to be structurally very similar. In fact, if rigid templates were used to match the heads, there may often be better alignment (lower match error) between the same views of the different heads (e.g. frontal) than between different views of the same head (e.g. frontal and three-quarter face). This emphasizes the importance of using flexible, non-rigid models, and matching methods for the purpose of identification, as was accomplished here.

In the discrimination experiment, the model image of every object was matched to every frame of every sequence described in the previous section. For each pair of model and current input images, the number of accepted peak matches was counted, and if the count was above some threshold, the modeled subject was considered identified in that input image. Since each model image had a different number of LoG peaks, a threshold was selected for each. Table 1 shows the thresholds and the resulting discrimination rates. The first column shows the matched peak count thresholds for each subject. The second column shows the percent of correct positives: the number of frames that were correctly identified as a given subject over the total number of frames of that subject. The third column shows the percent of false positives: the number of frames that were incorrectly identified as a given subject over the total number frames not containing that subject. The bottom row shows the averages for these statistics over all subjects studied; a total of 1,050 image pairs were tested.

Table 1: Discrimination results testing a total of 1,050 image pairs

Test subject	Required points	Correct positives	False positives
Subj A:	6	100%	0%
Subj B:	7	96%	3%
Subj C:	11	80%	4%
Average:	8	92%	2.3%

As table 1 shows, the system is capable of correctly identifying an object 92 percent of the time, and incorrectly identifies that object an average of only 2.3 percent of the time. This is against very similar objects, using only a single model frame, and a single current input frame. In addition, the Gaussian moment computation is performed only at a single scale, while the images are of varying scale.

6.5.3 Feasibility of Recognition

A recognition system based on Gaussian-weighted moments, flexible semi-local matching, and robust object localization has been motivated, described and demonstrated. Given its performance on realistic data when only one model image and one input image is used, it seems reasonable to conclude that a usable level of performance for security and surveillance applications can be achieved when integrated with the detection and tracking modules. When embedded in an integrated system, the subject identification process would be able to utilize multiple model images of the same subject, gathered as it is being tracked. Tracking of the object also facilitates matching of the model data to multiple frames of the current video sequence. This multiplicity of data would greatly enhance the recognition rates that are already very good. In addition, other modalities, such as color, can be used to further identify subjects. Color can be particularly useful for rapid re-acquisition of a subject that has been recently tracked: a person's clothing should be unchanged in that time. Such situations frequently occur in security and surveillance applications.

The recognition process can be efficiently implemented since it is constructed of a few components that are, in principal, very fast. The Gaussian and Laplacian filters have already been implemented to run in a fraction of a video frame period for the working version of the tracking system. The peaks of the Laplacian are simple to detect, and from that step on, the processing is restricted to pixels that are LoG peaks. This focuses the processing from tens of thousands of pixels to a few hundred. By using small peak neighborhoods only, and by employing indexing and voting, the rough matching step requires a minimum of computation time. This is followed by a local match analysis that runs only 5 cycles over a much reduced set of potential point matches. Given this, and given the fact that the tracked regions isolated for recognition analysis are only a fraction of the image, it should be possible to perform all this computation in a fraction of a second.

7 An elliptical head tracker

This section describes an approach for modeling visual objects for the purpose of guiding tracking, and more importantly, for recovering model parameters such as visual size for use in controlling camera zoom.

This modeling technique concentrates on a stable property of human subject's heads, namely, that their silhouettes are approximately oval in shape. As each new image becomes available, an elliptical model is used to determine the location of the head in the new image. This model is extremely crude, yet it works surprisingly well in some realistic, unmodified environments. Using this model, a standalone test system has been developed that tracks a person undergoing "normal" motions, such as walking around the room, turning around, sitting down, and standing up. The system actively controls the camera's pan and tilt to keep the subject centered in the field of view. Although the algorithm relies on a local search, a simple prediction technique, like that described in Section 3, on patch tracking, removes any restriction on maximum image velocity, thus allowing the subject to move at normal speeds. Because the algorithm does not use any properties inside of the face, it in insensitive to full 360-degree rotation of the body, as well as to small amounts of occlusion.

Section 7.1 gives an idea of the head tracker's usefulness. Then, the following three sections describe the algorithm in detail, including the motivation for using an ellipse, and various issues regarding camera control. Section 7.5 presents the tracker's performance in two different environments, which is followed by a section explaining a method to determine the tracker's confidence.

7.1 Research Goals

It is becoming increasingly apparent that no single visual technique is going to be able to provide robust, reliable tracking of a person in unmodified environments. Instead, a successful solution will probably be a well-integrated combination of two or more modules (see Figure 19 for an illustration of the general idea). Although our system has been used to successfully track a person for 30 seconds, or more, in some environments, the algorithm is not applicable to all environments, nor does it perform flawlessly even in the best of environments. Therefore, it alone is incapable of performing reliable, robust tracking. However, because of its simplicity and low overhead (in terms of computation), it seems to have potential as a module of a larger system. The architecture of such a system (that is, which modules are used, and how those modules interact and communicate) is beyond the scope of this section, and the interested reader is referred to the literature for proposals of this sort (e.g., [43]).

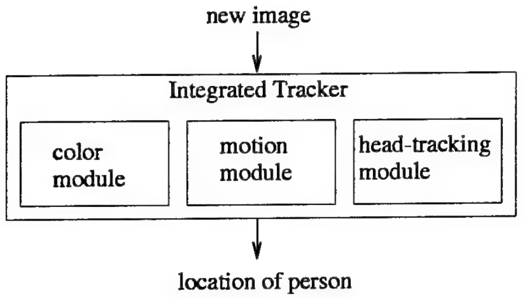


Figure 19: An example of a system using the ellipse-based head tracking module.

7.2 Head Model

A person's head has some interesting properties that make it an intriguing focus for tracking research. For example, the head has limited motion with respect to the torso, so knowledge of the head's location yields approximate knowledge of the rest of the body. Although more accurate localization of the body is desirable, knowledge of the head's location is often sufficient for camera servoing, since the head is arguably the most interesting part, and since most applications require the head to remain in the field of view (as opposed to the feet, for example, which may or may not be of interest). In addition, the head is easy to model, as the result of two geometrical properties. For one, it is nearly rigid, making rigid models appropriate much of the time. And secondly, it is fairly symmetric about the vertical axis passing through its center, which means that its 2-D projection onto the image plane is nearly constant.

As a result of this rigidity and symmetry, a rigid 2-D model goes a long way toward approximating the shape of the head in an image. The particular model that we use is an ellipse with a fixed aspect ratio of 1.2 and a fixed vertical orientation, as shown in Figure 20. This leaves three degrees of freedom: x location, y location, and size (i.e, length of the minor axis). In practice, we have found that the 1.2 aspect ratio is a valid approximation for many people's heads, and we show in Section 7.5 that the insistence on vertical orientation is not a limitation, since the ellipse correctly hangs on to the figure even when the head is turned sideways.

7.3 Finding the Head

The ellipse's state (i.e., its x location, y location, and size) is maintained by performing a local search each time a new image becomes available. The primary assumption (along with the head's elliptical shape) is that a relatively strong intensity gradient follows the contour of the head.

The algorithm proceeds as follows: The new image is smoothed by convolving with a 5×5 Gaussian filter. Then a rough approximation to the gradient magnitude is

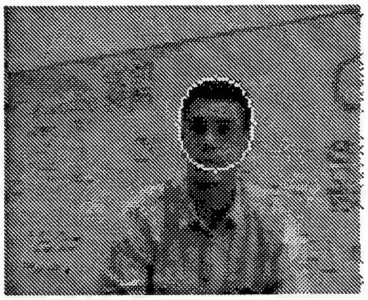


Figure 20: The elliptical head model.

obtained by convolving the image with a $[-1,0,1]$ horizontal filter, then convolving with a $[-1,0,1]^T$ vertical filter (the T denotes tranpose), and, finally summing the absolute values of the two results. We have found that thresholding the gradient so that any value above the threshold becomes saturated, while any value below the threshold remains unchanged, helps to reduce the sensitivity to contrast. Without this threshold, the search is sometimes attracted to really strong gradients on the background rather than mediocre gradients around the head. On the other hand, the algorithm seems to be insensitive to the actual value of this threshold.

Once the gradient is computed, a local, exhaustive search determines the best state (x^*, y^*, s^*) of the ellipse. "Best" is defined by maximizing the sum of the gradient around the perimeter of the ellipse, divided by the number of pixels that constitute the perimeter. More precisely,

$$(x^*, y^*, s^*) = \arg\max_{x,y,s} \left\{ \frac{1}{N_s} \sum_{i=1}^{N_s} G(x + x_{s_i}, y + y_{s_i}) \right\},$$

where $G(x, y)$ is the gradient at pixel (x, y), N_s is the number of pixels on the perimeter of an ellipse of size s, and $(x_{s_1}, y_{s_1}), \ldots, (x_{s_{N_s}}, y_{s_{N_s}})$ are the pixels on the perimeter, given with respect to the ellipse center. (These perimeter pixels are precomputed for the various sizes, to decrease processing time.) In our implementation, this search considers 867 possibilites, that is, 289 locations (± 8 pixels in x and y) and 3 sizes (± 1 pixel for the length of the minor axis).

Around what state does this local search begin? Rather than assuming that the subject is roughly stationary, we make the more reasonable assumption that the subject is moving at a nearly constant speed. Therefore, instead of starting the search around the previous location, the search is started around the predicted location given by adding a velocity vector to the previous location.[2] This velocity vector is obtained in a straightforward manner by subtracting the location in the image preceding the previous image from the location in the previous image.

This prediction scheme is trivial to implement, but improves the performance of the

[2] In our experiments, we have not found it necessary to predict the size of the ellipse; therefore, the search *does* begin around the previous size.

tracker substantially because it removes any restriction on maximum image velocity. Without prediction, the tracker cannot cope with normal speeds, which we have found to be up to 14 pixels per frame. Simply increasing the search range is not an adequate solution for two reasons: computing time increases with the square of the range, and, more importantly, a large search area increases the likelihood that the tracker will become confused by gradients in the background. Prediction completely removes the restriction on maximum velocity and replaces it with a restriction on maximum acceleration. This shift in emphasis allows the tracker to handle any motion, since the acceleration of the head is bounded in our system by about 8 pixels;[3] no matter how fast the subject shakes his head back and forth, he cannot lose the tracker.

Initialization of the tracker is a simple matter. The ellipse begins in the center of the image at maximum size, and the subject is required to position his head in the center also. When the tracking button is pressed, the ellipse quickly collapses and locks on to the head. We have found that the ellipse is much more able to collapse than to grow, because of gradients inside of the head. Although this procedure requires manual positioning of the head, the tracker also shows a remarkable ability to automatically reaquire the head when it is brought close to the ellipse, a feature that is demonstrated in Section 7.5.

7.4 Controlling the Camera

The experimental system used a Canon VC-C1 camera, which accepts velocity commands for the pan, tilt, and zoom motors. Only one motor can be actuated at a time. Although there seems to be a roughly 90 millisecond delay between the sending of a command and the response of a motor, we have found this delay to be of little consequence.

Because the camera accepts velocity commands, control is rather simple. The location of the ellipse is compared with the center of the image, and either a pan or a tilt command is sent, depending on whether the error in the x direction or the y direction is greater, respectively. The magnitude of the velocity is directly proportional to the error used. If both errors are below some threshold (around 10 pixels), then a **STOP** command is sent, to prevent unnecessary oscillation. Our system is unable to control zoom reliably, therefore the zoom motor remains stationary.

It is worth noting the advantage of a velocity-controlled camera over a position-controlled camera for this type of task. Woodfill and Zabih [44], for example, have reported some tedious complications that result from a position-controlled camera. In their system, the camera had to provide synchronized position feedback (in this case, the pan angle) with every frame. When the subject's location in the new image was determined, it had to be translated into a camera pan angle (which required calibration). This angle was then combined with the position recorded when the image was taken to

[3] Of course, this number is dependent on the focal length, distance to the subject, and pixel size.

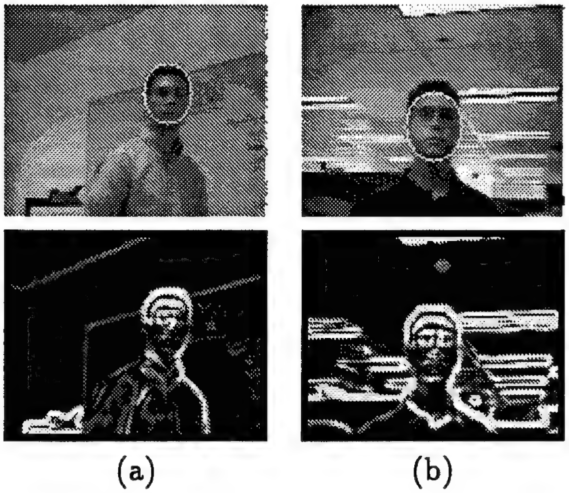


Figure 21: The two environments: (a) the "untextured" room, and (b) the "textured" room.

provide a desired angle with respect to a fixed home position. This final angle was the command sent to the camera.

7.5 Experimental Results

The tracker was tested in two different environments (shown in Figure 21). The first environment, known hence as the "untextured" room, consisted of a whiteboard flanked on one side by an overhead projector, and on the other side by a window with a vertical black border. Although the overhead and border caused large gradients in the image (see, for example, the lower-left corner of Figure 21a), the whiteboard caused only weak gradients. However, this does not mean that the whiteboard was clean—in fact writing filled the board. Since the markers were thin compared to the size of the head, their influence on the gradient was reduced by the Gaussian smoothing. From Figure 21a, it is not hard to see why the tracker worked so well in this environment. In contrast, the second environment, known as the "textured" room, contained lights on the ceiling that caused strong gradients in the image. Yet the tracker performed well here, too, since the gradients were not elliptically shaped.

In each environment, the subject was tracked for 1000 frames (about 30 seconds). During this time, the subject performed "normal" motions, such as walking, turning around, sitting down, and standing up. Figure 24 shows the tracker's performance in various situations. Sequences (a) through (e) were taken in the untextured room, while sequence (f) was taken in the textured room. The following is an examination of the sequences in detail:

(a). *Occlusion.* The tracker can handle occlusion, as long as neither the occluding object nor the background looks like an ellipse. In this example, the tracker maintained its fixture on the subject's head while the subject waved his arm in front. At one point the arm nearly completely covered the head, yet the tracker did not get lost. However, the tracker usually drifts from the head if both arms are waved simultaneously, because the crossing of the arms creates a cavity that looks like an ellipse.

(b). *Rotation.* One common failure mode for template-based trackers is rotation about an axis parallel to the image plane. Such a rotation causes parts of the object to disappear and other parts to reappear, which implies that a new template must be used. In contrast, the head tracker is not bothered at all by rotation, since it is tied to the outline of the object rather than the surface of the object. This example shows the ability of the tracker to follow the head even under 360-degree rotation, which causes the entire surface within the ellipse to disappear and then reappear. (Notice that the subject was walking as he was turning.)

(c). *Sitting.* This sequence demonstrates that the tracker was controlling the tilt of the camera, as well as its pan.

(d). *Reaquisition.* We have noticed an uncanny ability of the tracker to reaquire the subject when he returns to the camera's field of view. In this sequence, the ellipse had been stuck at the junction between the vertical window border and the ceiling for an extended period of time. When the subject reappeared and walked across the image, the ellipse slid down to lock onto his head. This behavior is more remarkable when one considers that, at the time of the reaquisition, the center of the subject's head was still 29 pixels away from the ellipse's original center. (Recall that the ellipse's search range is only ±8 pixels.) The fact that the top-right section of the head came within 8 pixels of the top-left section of the ellipse's perimeter was enough to attract the ellipse to the head.

(e). *Scaling.* As the subject walked closer to the camera, the size of the ellipse grew. (The excuse for cropping the subject's head in the last frame is that the camera's tilt joint limit had been reached.) It must be noted, however, that it is difficult to repeat this behavior. Typically when the subject walks toward the camera, the ellipse is attracted to gradients within the head, thus it either remains the same size or shrinks. Because the gradient around the chin tends to be weak, the ellipse often prefers the region around the nose (see the last image of Figure 24f) or the hair. As a result, we have found our headtracker to be unable to control zoom reliably.

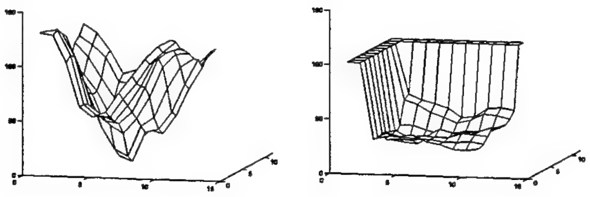
(f). *Textured background.* This sequence shows that the subject was successfully tracked in a more textured environment. Once again, 360-degree rotation posed no problem, although we lacked the space to show it here.

Another interesting behavior is shown in the last frame of Figure 24f. Although the orientation of the ellipse was cemented as vertical, the tracker was not confused when the subject tilted his head sideways. It is true that instead of properly outlining the head, the ellipse aligned itself with gradients in the interior (such as those around the nose), but the ellipse remained glued to the head nonetheless. This behavior was observed numerous times as the head was tilted 90 degrees to one side, then 90 degrees to the other side, then as the subject looked up toward the ceiling and down toward the floor. The tracker was not confused by any of these motions.

A few comments about speed are in order. In Section 7.3 the tracker was shown to handle arbitrarily large velocities, since its prediction scheme transforms the usual velocity limit to an acceleration limit. Indeed, in one experiment the subject tossed his head back and forth as fast as possible, and the tracker retained its fixation. However, two facts about the untextured sequence shown in Figure 24 show that the subject's speed was limited. For one, the tracker lost the subject about 2/3 of the way into the sequence because the subject walked too fast (Figure 23). Secondly, the subject was required to slow his movements when he sat down and stood back up again. These two problems are thought to be caused by poor camera control and response.

In addition to the sequences shown here, the tracker was tested about 50 times over the course of three months. Empirically, we found the tracker to be very successful in untextured environments with subjects whose hair color was different from the background. The tracker was less successful with balding subjects of light complexion, whose head outline did not contain strong enough gradients. Surprisingly, the tracker worked fine on a woman who had long black hair; in this case the ellipse enlarged itself to trace the outline of her hair rather than her face. In textured backgrounds, however, the results have been less promising. As long as the gradients were not elliptically shaped (as in the "textured" room), then the gradients were not too distracting to the tracker. However, in a room full of stacked cardboard boxes, the tracker failed miserably because the corners of the boxes yielded strong gradients that were roughly elliptical. It is clear that a general robust, reliable tracker will need to augment its awareness of elliptical gradients with other percepts.

To complete the discussion of experimental results, the details of the implementation are now given. The algorithm was implemented on a Hewlett-Packard personal computer equipped with a 133 MHz Pentium microprocessor. The Canon VC-C1 camera, coupled with a Matrox Meteor frame grabber, supplied the algorithm with 128 × 96 images every 33 milliseconds. It is difficult to accurately measure the amount of time taken up by the computation, but this description should help to give an idea: every 33 milliseconds, a new image was grabbed, the algorithm determined the new location of the ellipse, a command was sent to move the camera, and the new image was displayed on the screen. (The last two steps were relatively time-consuming.) We estimate that the algorithm itself takes between 12 and 18 milliseconds.



(a) (b)

Figure 22: The error function when (a) the ellipse is on the subject's head, and (b) it is in the process of losing the subject. The plots show xi versus x and y for one particular choice of s.

7.6 Measuring Confidence

One important property of a module embedded in a larger system is its ability to determine confidence in its output. Since the module will not work all the time (otherwise, there would be no need for a larger system), it must give an indication of the trustworthiness of its output, so that other modules can take over when necessary. The head tracker measures its confidence by looking at the shapes of its error functions: If the peaks are sharp, then the result of the search is trustworthy; otherwise it is not.

Recall from Section 7.3 that the search is conducted along the x and y dimensions, at three different scales (or sizes). For each state (x, y, s), a value is computed indicating the likelihood of that state. Although tracking discards all the values except the global maximum, confidence is measured by examining all the values. Three error surfaces, one at each scale, are computed by flipping the values upside down (so that the best state yields a global minimum rather than a global maximum). Then the curvature of each surface is approximated by counting the number of values that are at least a certain distance (in terms of value) away from the global minimum.[4] For example, if the minimum is shallow, then most of the values will be approximately the same, and few values will be very different from the minimum, yielding a low confidence. On the other hand, a sharp minimum will yield many values that are far from the minimum, yielding a high confidence. The confidences of the three surfaces are averaged to produce a single confidence.

Figure 22 illustrates the idea by displaying the middle scale's error function in two situations. On the left, the tracker is locked onto the head in an unambiguous environment, so the error function has a sharp minimum and a high confidence. In contrast, on the right, the tracker is about to drift from the head to some gradients on the background. In this case the error function is flat, which yields a low confidence.[5]

[4] The reader can probably suggest a better technique for estimating curvature.

[5] The large values surrounding two sides of the function signify that these values cannot be computed because of the ellipse's close proximity to the image boundaries.

The time history of confidences for the two 1000-frame sequences of the previous section are shown in Figure 23. The subject was successfully tracked throughout the first sequence, which was taken in the textured room. Notice that the confidences remain high, around 92 percent. In the second sequence, which was taken in the untextured room, there is a dip in the confidence around frame 560 even though the subject is being tracked. This is partly because the subject's head has become smaller than the smallest allowable ellipse, leading to an unstable location (see the last two frames of Figure 24c). About 100 frames later, around frame 650, the tracker loses the subject, because of the subject's high velocity. While the tracker is drifting from the subject's head to the background, the confidence drops to 50 percent, giving a good indication that the tracker is becoming lost. Once the tracker has settled onto a good location of the background, however, the confidence returns to a stable 84 percent. When the subject is reaquired around frame 930, the confidence increases only slightly to about 92 percent. Therefore, although a low confidence seems to indicate that the tracker might be losing the subject, a high confidence is only a good indicator if it is certain that the subject has not already been lost.

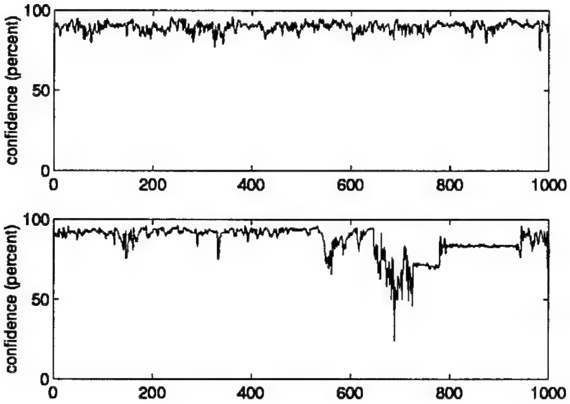
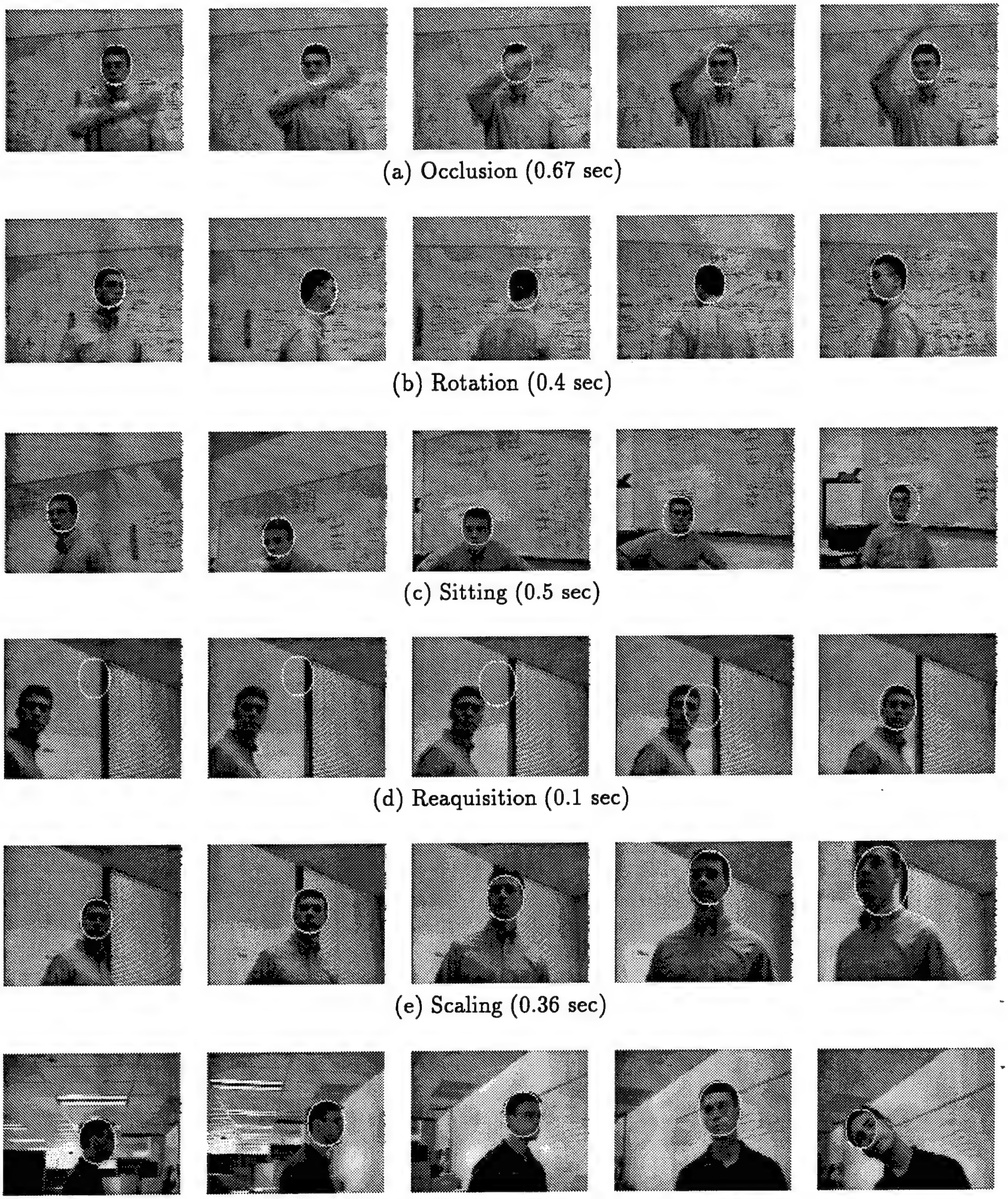


Figure 23: The confidence of the head tracker. Top: the subject is successfully tracked. Bottom: the subject is lost around frame 650 and reacquired around frame 930.

7.7 Summary of ellipse tracker results

This section describes experiments using a simple head modeling technique that has been used to follow a person in an unstructured environment. This modeling algorithm overcomes several common problems, including full body rotation and limited processor resources. However, the algorithm is heavily dependent on the single assumption that intensity gradients outline the subject's head. (Note that other algorithms also exhibit this dependence [45, 46].) Therefore, it is too fragile to be used alone and must be augmented by other techniques in order to provide a robust, general tracking system. Its low overhead and ability to measure confidence make it a viable candidate as one module in such a system.



(a) Occlusion (0.67 sec)

(b) Rotation (0.4 sec)

(c) Sitting (0.5 sec)

(d) Reaquisition (0.1 sec)

(e) Scaling (0.36 sec)

(f) Textured background (0.67 sec)

Figure 24: Demonstration of the head tracker's performance under various conditions. The number in parentheses indicates the amount of elapsed time between the displayed frames.

Several extensions could be made to the algorithm presented here. For example, instead of summing the magnitude of the gradient along the perimeter of the ellipse, the dot products of the gradient vector and ellipse normal could be summed. This would make the computation less sensitive to strong gradients in the background. Another possibility would be to relax the rigidity constraint by allowing the ellipse to deform to match the actual contour of the head. Finally, the contours of other body parts (such as the torso) could be tracked to give a more knowledgeable and robust representation of the subject's location.

8 Hardware/Software Performance Trends

Up until the last year or two, special purpose hardware accelerators have been required to carry out the massive image processing computations required for carrying out the image processing associated with the algorithms described in previous sections of this report.

Real-time vision processing speeds are limited by data movement bottlenecks and by available computation resources. Dedicated accelerators, such as Teleos' Prism-3 stereo and motion system[2, 47, 48] or JPL's DataCube stereo system[49], have been able to maintain high data throughput rates from digitizer to parallel processing pipelines, allowing intensive early vision computations to be carried out at video or near video rates.

However, the early advantage held by special board-level hardware over general purpose computers has been eroding over the past decade. Figure 25 illustrates the trends for a few examples of software and hardware stereo correlation systems. The figure compares representative systems by the number of correlations per second achieved by each. Among pure hardware systems, Prism-2[50] was an early instance that used conventional logic, such as adder and RAM chips, to implement a large kernel convolver and area correlator. Prism-3 used a similar architecture with more advanced Field Programmable Gate Array technology. This design ran at 4 times the clock rate and yielded almost an order of magnitude improvement in correlation speed. More recently CMU[51] developed a large piece of hardware that boosts performance by another two orders of magnitude. These data points suggest that hardware stereo correlators have been gaining about a factor of two in speed each year over the past decade.

Software stereo correlators on standard workstations, over the same period, have gone from being nearly 4 orders of magnitude slower [8] to just 1.5 orders of magnitude currently. If these trends persist for the remainder of the decade, software on personal computers will run essentially as fast as elaborate dedicated hardware systems by the turn of the century. There are several factors that give credibility to this somewhat paradoxical situation. First and most significantly, clock speeds for board level accelerator designs have only risen by a factor of 4 or so in 10 years because of the physical limitations of clocking data onto pieces of wire. At the same time the instruction rate on microprocessors has risen by 3 orders of magnitude during the same period. Closely coupled to this difference is the fact that investments made in making commodity processors faster far outpaces what can be justified for very low volume hardware accelerators for motion and stereo correlation. Another important factor comes from improvements in processor bus bandwidth driven by the multimedia revolution. The current PCI bus on personal computers is specified to allow as many as 10 live video signals to be moved simultaneously from video sources to host memory. This eliminates a critical bottleneck faced by earlier software implementations. Finally, most of the design techniques that have benefited hardware implementations, such as exploitation of separable convolutions,

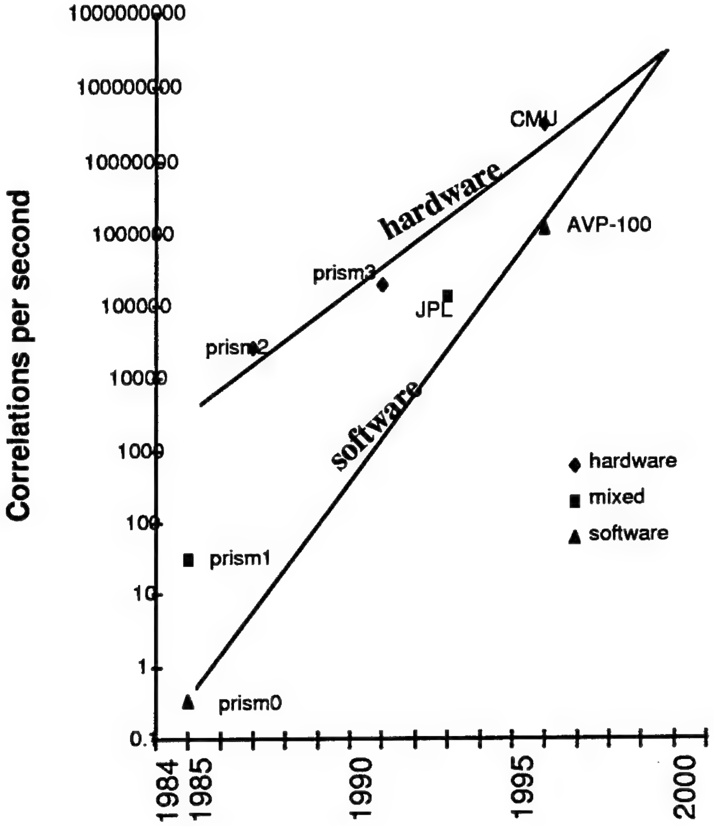


Figure 25: Software versus hardware performance trends for stereo correlation. Measurement rates for a representative set of stereo correlation systems developed over the past ten years are compared. The plot suggests that the performance gap between hardware- and software-based systems is closing steadily.

binomial approximations to the Gaussian, and boxcar filters, also have helped software implementations.

Based on these hardware versus software trends, Teleos Research has directed its technology development efforts toward PC-based software systems for early visual processing. The results have been favorable and commercial sensing products are now on the market which carry out real-time tracking of people for teleconferencing, distance learning, and security applications. This tracking technology makes use of the same convolution and correlation algorithms employed under the DARPA Unmanned Ground Vehicle Demo II program for stereo range finding.

As an example, an outdated 66 MHz Pentium PC runs the motion-based figure detection algorithm described in Section 2.2.1 at video rate, and drives an active camera head to follow subjects walking in an office environment. A 133 MHz dual Pentium system is capable of runing the same tracking software and computing stereo range images in parallel. As processor performance increases from year to year, this same software will be able to run at increasingly higher resolutions and additional analysis modules will be able to run concurrently without affecting the low-level vision processing.

It is anticipated that multimedia enhancements expected in the next generation of Pentium processors will enable another quantum leap in visual measurement performance over the linear clock speed trend line.

As we move past the software versus hardware crossover point discussed above, the possibility of exploiting the opposite extreme of the big versus small system dimension arises. Once a small system can exceed a minimum performance level, a snowball effect the other way can occur. Cheaper systems do not have to do as much to be useful, and since they are more expendable they do not have to operate as conservatively which makes it easier to build them more cheaply.

As noted above, even commodity laptop PC's have sufficient power to carry out real-time tracking; similarly, commodity frame grabbers and cameras developed for the multimedia market are adequate applications in the S&S task domain.

9 Technology transfer

Teleos is actively working to transfer its RTPC-sponsored technology into practical commercial applications. Collaborations with several major companies in the security and video teleconferencing areas are underway.

9.1 Scientific interaction

Results of this research effort were shared with members of the Real-Time Planning and Control Program at PI meetings and at technical conferences.

Teleos Research also prepared and operated a live demonstration of Real-time Planning and Control (RTPC) funded research results at DARPA's SISTO symposium, held in the Washington D.C. area September 26 through 31, 1995. Demonstrations also were carried out at DARPA Image Understanding Workshops and at Unmanned Ground Vehicle Demo II meetings. These demonstrations often highlighted collaborative work between Teleos and other DARPA contractors, bringing together results from several DARPA supported programs including RTPC, Image Understanding, and Unmanned Ground Vehicle.

RTPC-sponsored research results have been run as an application layer on top of the AVP-100 system, a low-cost, commercially available, real-time visual measurement platform. This system, including figure-tracking algorithms, has been delivered to six other research laboratories as of the date of this report.

9.2 Potential Post Applications

Teleos sees an opportunity to meet the increasing demand for systems that take advantage of online sensor data, especially real-time video data using technology developed under this research contract. The effect of this new technology will be to extend the range of economically feasible applications and to accelerate the cost reduction of visual sensing components. Several of the applications that would benefit from these developments are described in this section.

9.2.1 Applications of Interest to the Federal Government

Security and Surveillance. In the security and surveillance domain, the ability to do quick detection and classification of objects can add value both to aerial reconnaissance and to ground-based systems monitoring an interior area for intruders or performing outdoor perimeter control. Current-generation motion-detection systems are hampered by their inability to recognize the objects causing the motion, e.g., to distinguish the motions of a cat from those of a human. Recognition techniques to be developed under this project would immediately increase the value of intrusion-detection systems and reduce the personnel needed to man them.

Material-Handling Applications. The following high-payoff material-handling applications have been described to Teleos in its discussions with technology representatives of organizations within the government:

NASA: Satellite grappling

Army, Air Force: Armaments handling

Navy: Object tracking for online maintenance

Marine Corps: Supply unloading in forward landing zones

Postal Service: Package handling, address block location

In each of these applications, specialized robotic equipment is being developed to reduce manpower requirements and to enhance the safety and effectiveness of human personnel. The robustness of the control software that guides this equipment, however, is currently limited by the speed and quality of perceptual data. Typically, specialized sensors are employed to take the precise measurements required for inspection and positioning. Because these sensors are so specialized, however, system designers are ordinarily forced to make very strong assumptions about the overall situation (e.g., that the robot is indeed facing the object of interest, or that it is in approximately the right position and orientation.) These assumptions lead to brittleness in the control software. Technology for fast, fluent recognition would provide designers with just the modular component needed to build systems with a broader domain of applicability and more robust performance.

Intelligent Traffic Monitoring. In the area of transportation, intelligent vehicle monitoring has been receiving increased attention during the last several years. Tracking systems are currently being developed for following the motions of individual vehicles on the highway, and for identifying whether traffic is flowing smoothly, whether a car is stalled, driving erratically, and so on. These techniques tend to use fairly simplistic criteria, however, for deciding where a vehicle is, if it is indeed a vehicle, and what it is doing. The competence and robustness of these systems could be increased dramatically through improved real-time recognition.

9.2.2 Applications of Interest to the Private Sector

Security and Surveillance. The benefits in this application area mentioned under government uses apply equally to the civilian domain. As a follow-on to its government-funded work in intelligent security monitoring, Teleos Research has entered into a relationship with a leading private-sector manufacturer of security systems with the aim of incorporating intelligent tracking as a feature in next-generation systems. Preliminary discussions with other parties in the security industry confirm a high degree of interest in providing this type of capability.

Entertainment. Similarly, preliminary contacts with representatives of the entertainment industry indicate a strong interest in adding real-time recognition and tracking to interactive games and animatronic displays. The purpose of visual tracking in such systems would be to heighten the interactive experience for the user by making the system more aware of the player and able to respond to his identity and movements.

3D Modeling. A variety of applications in the 3-D modeling area would also benefit from real-time recognition and tracking of objects. Most 3-D models are currently developed by entering the models manually, rather than through direct sensing. Some systems exist for doing 3-D scanning, but these are expensive and difficult to set up and operate. A more natural way to acquire 3-D models would be to derive them directly from video imagery, either from binocular stereo inputs, or from monocular structure from motion. Real-time object detection and tracking could be an important component in this process.

Video Communications. Video teleconferencing is another area of high commercial interest that would benefit from automated recognition and tracking. Communication bandwidth is at a premium in video communications, hence, it is extremely desirable to be able automatically to direct a camera to a face or another object, and to maintain high pixel resolution over the areas of interest. For this reason, real-time figure isolation, classification, and tracking would provide a valuable component in an intelligent videoconferencing system.

10 Summary

This report details progress that Teleos has made in the development of computer vision and visual attention mechanisms for the support of a S&S-directed vision and planning system.

We have presented theories and experimental results in five related areas. Figure-ground discrimination, tracking servos, object recognition based on parts extraction, object recognition based on local geometric structure, and object modeling were obtained. This work has fostered the development of enhanced visual perception capabilities relevant to security and surveillance detection and tracking. We also have discussed trends toward real-time software implementations on commodity processors which will make S&S tracking technology practical for a wide range of applications.

References

[1] Nishihara, H. K. Minimal meaningful measurement tools. Technical Report TR-91-01, Teleos Research, 1991.

[2] Nishihara, H. K., H. Thomas, and E. Huber. Real-time tracking of people using stereo and motion. In *Machine Vision Applications in Industrial Inspection II, Benjamin M. Dawson, Stephen S. Wilson, Frederick Y. Wu Editors*, volume Proc. SPIE 2183, pages 266–273, 1994.

[3] Huttenlocher, D. P., J. J. Noh, and W. J. Rucklidge. Tracking non-rigid objects in complex scenes. In *Proceedings of the Fourth International Conference on Computer Vision*, pages 93–101, Berlin, Germany, May 11-14, 1993.

[4] Wren, C. et al. Pfinder: Real-time tracking of the human body. In *Proceedings of the SPIE*, volume 2615, pages 89–98, 1996.

[5] Murray, D. and A. Basu. Motion tracking with an active camera. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(5):449–459, May 1994.

[6] Swain, M. J. Color indexing. Technical Report 360, Computer Science Department, University of Rochester, NY, November 1990.

[7] Hunke, M. and A. Waibel. Face locating and tracking for human-computer interaction. In *Proceedings of the 28th Asilomar Conference on Signals, Systems and Computers*, pages 1277–1281, 1994.

[8] Nishihara, H. K. Practical real-time imaging stereo matcher. *Optical Engineering*, 23(5):536–545, October 1984. Also in *Readings in Computer Vision: Issues, Problems, Principles, and Paradigms*, edited by M.A. Fischler and O. Firschein, Morgan Kaufmann, Los Altos, 1987.

[9] Woodfill, J. and R. Zabih. An algorithm for real-time tracking of non-rigid objects. In *Proceedings of the 9th National Conference on Artificial Intelligence(AAAI-91)*, pages 718–723, 1991.

[10] Huttenlocher, D. P. and S. Ullman. Object recognition using alignment. In *Proceedings of the First International Conference on Computer Vision*, pages 102–111, 1987.

[11] Burns, J. B. and S. J. Rosenschein. Recognition via blob representation and relational voting. In *Proc. IEEE Conf. on Signals, Systems and Computers*, November 1993.

[12] Marr, D. and H. K. Nishihara. Representation and recognition of the spatial organisation of 3-D shapes. *Proceedings of the Royal Society of London*, 200:269–294, 1978.

[13] Beymer, D. Face recognition under varying pose. Artificial Intelligence Lab. Technical Report 1461, Massachusetts Institute of Technology, Cambridge, MA, 1993.

[14] Bichsel, M. and A. Pentland. Human face recognition and the face image set topology. In *CVGIP: Image Understanding*, volume 59, pages 254–261, March 1994.

[15] Blum, H. Biological shape and visual science, part 1. *J. Theoretical Biology*, 38:205–287, 1973.

[16] Blum, H. and R. N. Nagel. Shape description using weighted symmetric axis figures. *Patt. Rec.*, 10:167–180, 1978.

[17] Dill, A., M. Levine, and P. Noble. Multiple resolution skeletons. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(4):495–504, July 1987.

[18] Pizer, S., W. Oliver, and S. Bloomberg. Hierarchical shape description via the multiresolution symmetric axis transform. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(4):505–511, July 1987.

[19] Nishihara, H. K. A system for recognizing the shape of printed words. In *AAAI Spring Symposium*, March 1988.

[20] Crowley, J. L. and A. C. Parker. A representation for shape based on peaks and ridges in the difference of low-pass transform. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, PAMI-6(2):156–170, 1984.

[21] Canny, J. F. A computational approach to edge detection. *IEEE Transactions on Pattern Matching and Machine Intelligence*, 8(6):679–698, 1986.

[22] Witkin, A. Scale-space filtering. In *Proceedings of the Seventh International Joint Conference on Artificial Intelligence*, pages 1019–1022, Karlsruhe, West Germany, 1983.

[23] Subirana-Vilanova, J. and K. Sung. Perceptual organization without edges. In *Proceedings of DARPA Image Understanding Workshop*, pages 289–298, Jan. 1992.

[24] Gauch, J. and S. Pizer. Multiresolution analysis of ridges and valleys in grey-scale images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(6):635–645, June 1993.

[25] Koenderink, J. The structure of images. *Biol. Cyber.*, 50:363–370, 1984.

[26] Lindeberg, T. Detecting salient blob-like image structures and their scales. *IJCV*, 11(3):283–318, 1993.

[27] Morrone, M. and R. Owens. Feature detection from local energy. *Pattern Recognition Letters*, 6:303–313, 1987.

[28] Thompson, D. W. and J. Mundy. Three-dimensional model matching from an unconstrained viewpoint. In *Proceedings IEEE International Conference on Robotics and Automation*, pages 208–220, Raleigh, NC, 1987.

[29] Lamdan, Y. and H. Wolfson. Geometric hashing: A general and efficient model-based recognition scheme. In *Proceedings of the Second International Conference on Computer Vision*, pages 238–249, Tampa, FL, December 5-8, 1988.

[30] Califano, A. and R. Mohan. Systematic design of indexing strategies for object recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, page 709, New York, NY, June 15-17, 1993.

[31] Khotanzad, A. and Y. Hong. Invariant image recognition by zernike moments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(5), May 1990.

[32] Reiss, T. *Recognizing planar objects using invariant image features.* Springer-Velag, 1993.

[33] Ballard, D. and L. Wixson. Object recognition using steerable filters at multiple scales. In *IEEE Workshop on Qualitative Vision*, page 2, 1993.

[34] Rao, R. and D. Ballard. Object indexing using iconic sparse distributed memory. In *Proceedings of the Fifth International Conference on Computer Vision*, page 24, Cambridge, MA, 1995.

[35] Leung, T., M. Burl, and P. Perona. Finding faces in cluttered scenes using random labeled graph matching. In *Proceedings of the Fifth International Conference on Computer Vision*, page 637, Cambridge, MA, 1995.

[36] Wu, X. and B. Bhanu. Gabor wavelets for 3-D object recognition. In *Proceedings of the Fifth International Conference on Computer Vision*, page 537, Cambridge, MA, 1995.

[37] Freeman, W. and E. Adelson. The design and use of steerable filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(9):891, Sept. 1991.

[38] Manmatha, R. and J. Oliensis. Extracting affine deformations from image patches. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, page 754, New York, NY, June 15-17, 1993.

[39] Lindeburg, T. and J. Garding. Shape from texture from a multi-scale perspective. In *Proceedings of the Fourth International Conference on Computer Vision*, page 683, Berlin, Germany, May 11-14, 1993.

[40] Hager, G. D. and P. N. Belhumeur. Real-time tracking of image regions with changes in geometry and illumination. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 403–410, San Francisco, CA, 1996.

[41] Kumar, R. and A. Hanson. Robust methods for estimating pose and a sensitivity analysis. In *CVGIP: Image Understanding*, pages 313–342, 1994.

[42] Olson, C. On the speed and accuracy of object recognition when using imperfect grouping. In *International Symposium on Computer Vision*, pages 449–454, 1995.

[43] Toyama, K. and G. D. Hager. Incremental focus of attention for robust visual tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 189–195, San Francisco, CA, 1996.

[44] Woodfill, J. and R. Zabih. Using motion vision for a simple robotic task. In *Proceedings of the AAAI Fall Symposium of Sensory Aspects of Robotic Intelligence*, 1991.

[45] Blake, A., R. Curwen, and A. Zisserman. Affine-invariant contour tracking with automatic control of spatiotemporal scale. In *Proceedings of the Fourth International Conference on Computer Vision*, pages 66–75, Berlin, Germany, May 11-14, 1993.

[46] Lowe, D. G. Robust model-based motion tracking through the integration of search and estimation. *International Journal of Computer Vision*, 8(2):113–122, Aug. 1992.

[47] Nishihara, H. K. and S. J. Rosenschein. Stereo machine vision in fpga's. In *PLD Con95*, pages 25–27, Santa Clara, CA, April 1995. CMP Publications Inc, Manhasset NY.

[48] Marks, R., S. Rock, and M. Lee. Real-time video mosaicking of the ocean floor. In *Proceedings of IEEE Symposium on Autonomous Underwater Vehicle Technology*, Cambridge, MA, July 1994.

[49] Matthies, L. H. Stereo vision for planetary rovers: stochastic modeling to near real-time implementation. *International Journal of Computer Vision*, 8(1):71–91, 1992.

[50] Nishihara, H. K. Real-time implementation of a sign-correlation algorithm for image-matching. Technical Report TR-90-2, Teleos Research, 1990.

[51] Kimura, S., H. K. T. Kanade, A. Yoshida, E. Kawamura, and K. Oda. Video-rate stereo machine. In *Mobile Mapping Symposium*, Columbus, OH, May 1995.